Central Washington University
ScholarWorks@CWU

All Master's Theses | Master's Theses

Spring 2021

Visualization for Solving Non-image Problems and Saliency Mapping

Divya Chandrika Kalla
Central Washington University, divyachandrika.kalla@cwu.edu

Follow this and additional works at: https://digitalcommons.cwu.edu/etd

 Part of the Computational Engineering Commons, Computer Sciences Commons, and the Other Computer Engineering Commons

Recommended Citation

Kalla, Divya Chandrika, "Visualization for Solving Non-image Problems and Saliency Mapping" (2021). *All Master's Theses*. 1507.
https://digitalcommons.cwu.edu/etd/1507

This Thesis is brought to you for free and open access by the Master's Theses at ScholarWorks@CWU. It has been accepted for inclusion in All Master's Theses by an authorized administrator of ScholarWorks@CWU. For more information, please contact scholarworks@cwu.edu.

VISUALIZATION FOR SOLVING NON-IMAGE PROBLEMS AND SALIENCY MAPPING

A Thesis

Presented to

The Graduate Faculty

Central Washington University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

Computational Science

---

by

Divya Chandrika Kalla

June 2021

CENTRAL WASHINGTON UNIVERSITY

Graduate Studies

We hereby approve the thesis of

Divya Chandrika Kalla

Candidate for the degree of Master of Science

APPROVED FOR THE GRADUATE FACULTY

_____	_____
	Dr. Boris Kovalerchuk
_____	_____
	Dr. Razvan Andonie
_____	_____
	Dr. Szilárd Vajda
_____	_____
	Dean of Graduate Studies

ABSTRACT

VISUALIZATION FOR SOLVING NON-IMAGE
PROBLEMS AND SALIENCY MAPPING

by

Divya Chandrika Kalla

June 2021

High-dimensional data play an important role in knowledge discovery and data science. Integration of visualization, visual analytics, machine learning (ML), and data mining (DM) are the key aspects of data science research for high-dimensional data. This thesis is to explore the efficiency of a new algorithm to convert non-images data into raster images by visualizing data using heatmap in the collocated paired coordinates (CPC). These images are called the CPC-R images and the algorithm that produces them is called the CPC-R algorithm. Powerful deep learning methods open an opportunity to solve non-image ML/DM problems by transforming non-image ML problems into image recognition. The main idea behind CPC-R is splitting attributes of an n-D point into consecutive pairs of its attributes, locating pairs in the same 2-D Cartesian space, and assigning grey scale intensities or colors to the pairs. There are several parameters that can be changed producing several versions of CPC-R images allowing to optimize images for classification. This thesis reports the results of computational experiments with the CPC-R algorithm for different Convolution Neural Network classifiers, and the methods to optimize the several versions of CPC-R images for the same n-point. These results show that the combined CPC-R and deep learning Convolution Neural Network algorithms are able to solve non-image Machine Learning problems reaching high accuracy on the benchmark datasets. The second part of this thesis reports the results

of Saliency Mapping with the CPC-R algorithm. The saliency models take an image and generate a saliency map that predicts which regions of the image will most likely draw a human viewer's attention. The saliency mappings with the CPC-R are explored, and further optimization studies are outlined. This thesis reports the importance of features by estimating the change of prediction accuracy due to the exclusion of the individual features. The large sets of pixels are used as features that can capture a large context. This approach views a cell as the most informative if covering it leads to the largest decrease in classification accuracy. This method is called the Informative Cell Covering (ICC) algorithm.

Keywords: Knowledge Discovery, Deep Learning, Collocated Paired Coordinates, Convolutional Neutral Networks, Raster Images, Machine Learning, Visualization, Non-image data, Data conversion.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Dr. Boris Kovalerchuk, Prof. of Computer Science at Central Washington University, for helping me in my entire thesis work.

I am extremely grateful that Prof Dr. Boris Kovalerchuk, agreed to be my thesis adviser and allowed me to work on a thesis that is related to data visualization and also for providing advice along with great ideas to achieve my goals. This thesis would not be completed without my advisor's passionate participation and inputs.

I would also like to thank committee members Prof. Dr. Razvan Andonie and Prof. Dr. Szilárd Vajda, who gave great feedback which helped during this work of the thesis.

I would like to thank my colleagues, friends, and family for listening, talk about my research, and for providing me with feedback. This accomplishment would not have been possible without them. Thank you.

Author

Divya Chandrika Kalla

TABLE OF CONTENTS

Chapter Page

Chapter Page

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

Table Page

Table Page

LIST OF TABLES (CONTINUED)

LIST OF FIGURES

Figure Page

Figure Page

LIST OF FIGURES (CONTINUED)

CHAPTER I

INTRODUCTION

Transforming Non-Image Machine Learning Problems Into ImageRecognition Problems

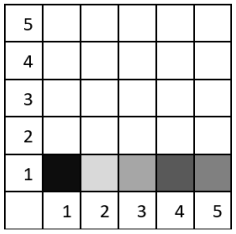
Important opportunities for data science research include the integration of visualization, visual analytics, machine learning, and data mining. Solving image recognition problems using deep learning algorithms has displayed advantages in several deep learning architectures for different types of images [1].

Visualization expands the knowledge discovery opportunities by converting non-image data to images for solving a variety of Machine Learning problems. In this way, a non-image classification task is converted into an image recognition task to be solved by powerful DL algorithms. To be successful with these visualizations, one needs to set up the criteria for efficiency and explore ways for effective visualizations.

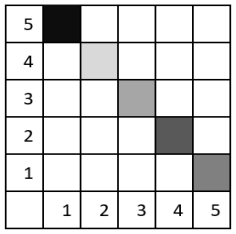
Single Value Mapping

In the single value mapping, a function *F* maps each value x_i of the n-D point *x* = $(x_1, x_2, \ldots, x_n)$ to a cell/pixel p_i with coordinates (p_{i1}, p_{i2}), $F(x_i) = (p_{i1}, p_{i2})$. The pixel p_i gets intensity equal to the coordinate value x_i that can be normalized, $I(p_i) = x_i$. This mapping is called single value mapping. The image size defines the mapping density [2]. The grey scale intensities are normalized into the [0,1] interval. See Fig. 1.

Example 1: $F(0)=(1,1)$, $F(0.7)=(1,2)$, $F(0.6)=(1,3)$, $F(0.2)=(1,4)$, $F(0.4)=(1,5)$.



Example 2: $F(0)=(1,5)$, $F(0.7)=(2,4)$, $F(0.6)=(3,3)$, $F(0.2)=(4,2)$, $F(0.4)=(5,1)$.



Example 3: $F(0)=(1,5)$, $F(0.7)=(2,5)$, $F(0.6)=(1,4)$, $F(0.2)=(2,4)$, $F(0.4)=(1,3)$.

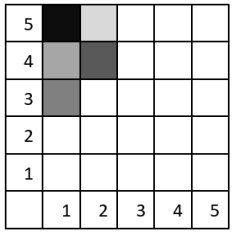


FIGURE 1: 5-D points in a single value mapping.

The disadvantage of this mapping is the size of its raster image because drawing graph edges requires many pixels. Both methods [3] use images that they produce as inputs to CNN algorithms for image recognition and both reported success on different benchmark data sets [4]. Their disadvantage is a single value mapping requiring a cell (pixel) for each x_i of n-D point x.

The proposed CPC-R is a new algorithm, that uses a pair values mapping with preserving n-D information. It requires two times less cells (pixels) than single value mappings. Its images are smaller than in [3] and with significantly less occlusion than in [3] by removing the graph edges.

Pair Value Mapping

In pair mapping, a function *H* is mapping a pair of values (x_i,x_j) to a single-pixel *p*, $H(x_i,x_j)$= (p_{1ij}, p_{2ij}). The function *H* is called pair value mapping.

In contrast with a single value mapping *H*, the values x_i and x_j are encoded by the location of the pixel (p_{i1}, p_{i2}). In the simple mapping, the coordinates of the pixels are x_i and x_j, (p_{1ij}, p_{2ij}) = (x_i,x_j) and ($p_{1i,i+1}, p_{2i,i+1}$) = (x_i, x_{i+1}). This can require rounding x_i and x_j to be integers. For the odd *n*, the last pair can be formed as (x_i,x_n) or (x_n,x_i) with any (x_i) including (x_n) itself. The grey scale intensity of the pixels are used to identify a pair, where the first pair is black; the second pair is dark grey, and so on until light grey. Alternatively, a color sequence can be used as in heatmaps. The values of intensities and colors can be optimized at the learning stage. The optimization of pairing was most beneficial to improve the accuracy.

The advantage of pair values mapping is that $[n/2]$ pixels can represent the n-D point without loss of information, i.e., two times less pixels than a single pixel mapping, which simplifies images. The images are smaller than in [5] and with significantly less occlusion than in [5] by removing the graph edges. The CPC-R optimizes the location of the pixels by allowing any pairs (x_i,x_j) not only (x_i, x_{i+1}) to improve classification accuracy. The simplest version generates randomly a fixed number of alternatives and maximizes accuracy on the training data, builds a model, and tests the validation data. This method is denoted as CPC-R.

The main part of this paper describes the experiments with the CPC-R and CNN algorithms to classify the real world and simulated data. The efficiency of the CPC-R+CNN algorithm is tested successfully by evaluating the accuracy of the classification of several CNN algorithms in solving learning problems, on images produced by CPC-R visualizations on benchmark data. For non-integer data, values are discretized to map to the CPC-R matrix.

The purpose of performing the all experiments is to improve the images for human visual analysis and improve the deep learning CNN algorithms results. Multiple experiments have been conducted to explore the impact of combinations of properties of images on the accuracy of classification.

Design of CPC-R Algorithm

The proposed **CPC-R** is a new algorithm, that uses pair values mapping with preserving n-D information. The CPC-R algorithm is used to convert the non-image data to images using Collocated Paired Coordinates (CPC) that have been modified for this purpose. The original CPC visualization approach was successfully applied to several real-world visual knowledge discovery tasks, such as UN world hunger data and developing a market trading strategy [6]. This thesis reports the successful testing efficiency of the CPC-R algorithm on benchmark datasets.

The main idea of CPC is splitting attributes of an n-D point $x = (x_1, x_2, ..., x_n)$ to consecutive pairs $(x_1, x_2), (x_3, x_4), ...(x_{n-1}, x_n)$. If *n* is an odd number, the last attribute is repeated to get n+1 attributes. Then all pairs are shown as 2-D points in the same 2-D Cartesian coordinate system and connected by arrows to form a directed graph: $(x_1, x_2) \rightarrow (x_3, x_4) \rightarrow ... \rightarrow (x_{n-1}, x_n)$.

This graph is equivalent to the n-D point *x* and can be fully restored from the graph. This property makes CPC a reversible lossless visual representation of the n-D point

alongside other General Line Coordinates (GLC) that have this property as well [6][3]. In GLC, *n* coordinates can be located in 2-D or 3-D in multiple ways including parallel radial, shifted, and collocated, under different angles. Next, coordinates can be straight or curvilinear.

The line L_i has the length of x_i of the n-D point, where each n-D point is presented as a sequence of straight lines connected under different angles. In [5], each n-D point from training and validation data is visualized as a separate image using the GLC-L approach that builds 2-D graphs without using heatmaps. Image recognition CNN algorithms successfully use these deep learning CNN algorithms with inputs from these images [5]. Using such deep learning algorithms one can solve a non-image classification problem by converting it to an image recognition task. However, the GLC-L approach requires compression of the image of the 2-D graph to make images smaller. The CPC-R approach is free from this deficiency because it does not need to visualize the graph's edges. The next chapter describes the versions of the CPC-R algorithm. The steps are well elaborated in versions of the CPC-R algorithm.

CHAPTER II

VERSIONS OF CPC-R ALGORITHM

CPC-R version 1.0

1. Split attributes of an n-D point $x = (x_1, x_2, ..., x_n)$ to consecutive pairs $(x_1, x_2), (x_3, x_4), ...(x_{n-1}, x_n)$. If *n* is odd repeat the last attribute of *x* to make *n* even.

2. Set up a cell size to locate pairs in the image. Each cell can consist of a single-pixel or dozens of pixels.

3. Locate pairs (x_i, x_{i+1}) in the image at the cell coordinate pair (x_i, x_{i+1}) in the image.

4. Assign the grey scale intensity to cells from black for (x_1, x_2) and very light grey for (x_{n-1}, x_n). Alternatively, assign heatmap colors.

5. Optional: Combine produced images with context information in the form of average images of classes.

6. Call/Run ML/DL algorithm to discover a predictive model.

7. Optional: Optimize intensities in S4 and pair coordinates beyond the sequential pairs (x_i, x_{i+1}) in S1 by the methods ranged from random generating a fixed number of alternatives to genetic algorithms with testing ML prediction accuracy on these alternatives.

Design of CPC-R Images with version 1.0

A CPC-R method is used to make the raster images and addresses the issue in the following way. It locates pairs (x_1, x_2) and uses the greyscale intensity from black for (x_1, x_2) and very light grey for (x_{n-1}, x_n) for nodes instead of connecting them by arrows. Alternatively, the intensity of a color is used. This order of intensities allows full restoration of the order of the pairs and their values from the image.

The size of the nodes (cells) can be varied from a single pixel to dozens of pixels. For instance, if each attribute has 10 different values then a small image with 10×10 pixels can represent a 17- D point by locating ten grey scale pixels in this image. This visualization will be lossless under the assumption that values of all pairs (x_i, x_{i+1}) are different and do not repeat. A treatment of colliding pairs is presented later. Generation of raster images, in CPC-R, is described in more detail and is illustrated in Fig. 2 for the Ionosphere Data [7]. In all experiments, we explore the impact of image sizes and origins of coordinates in the low left corner (LLC).

The idea of CPC-R images is illustrated with an example of a 17-D point **x**=(10, 5, 9, 4, 9, 5, 9, 3, 10, 5, 9, 4, 7, 2, 8, 3, 9, 3, 7, 3, 7, 3, 6, 2, 7, 2, 7, 2, 6, 3, 7, 2, 5, 2) from the Ionosphere data is converted to 17 pairs. Fig. 2 is based on the 17-D point.

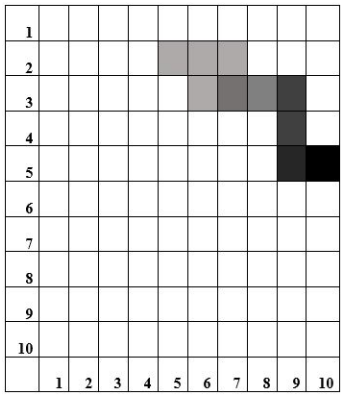


FIGURE 2: 17-D points in CPC-R.

The process of generation a CPC-R images for this case is as follows:

The consecutive pairs of values are formed from x = (10, 5, 9, 4, 9, 5, 9, 3, 10, 5, 9, 4, 7, 2, 8, 3, 9, 3, 7, 3, 7, 3, 6, 2, 7, 2, 7, 2, 6, 3, 7, 2, 5, 2):

(10,5), (9,4), (9,5), (9,3), (10,5), (9,4), (7,2), (8,3), (9,3), (7,3), (7,3), (6,2), (7,2), (7,2), (6,3), (7,2), (5,2).

The gray scale intensities are normalized from 0 to 1. The following steps are conducted:

- Filling cell (10,5) for the first pair (x_1, x_2) = (10,5) according to the grey scale value for the first pair set into 0 (black).
- Filling cell (9,4) for the second pair (x_3, x_4) = (9,4) according to the grey scale value for the second pair set into 0.2 (very dark grey).
- Filling cell (9,5) for the third pair (x_5, x_6) = (9,5) according to the grey scale value for the third pair set into 0.3 (very dark grey).
- Filling cell (9,3) for the fourth pair (x_7, x_8) = (9,3) according to the grey scale value for the fourth pair set into 0.35 (very dark grey).
- Filling cell (10,5) for the fifth pair (x_9, x_{10}) = (10,5) according to the grey scale value for the fifth pair set into 0.40 (very dark grey).
- Filling cell (9,4) for the sixth pair (x_{11}, x_{12}) = (9,4) according to the grey scale value for the sixth pair set into 0.45 (very dark grey).
- Filling cell (7,2) for the seventh pair (x_{13}, x_{14}) = (7,2) according to the grey scale value for the seventh pair set into 0.50 (very dark grey).
- Filling cell (8,3) for the eighth pair (x_{15}, x_{16}) = (8,3) according to the grey scale value for the eighth pair set into 0.55 (grey).

- Filling cell (9,3) for the ninth pair (x_{17}, x_{18}) = (9,3) according to the grey scale value for the ninth pair set into 0.60 (grey).

- Filling cell (7,3) for the tenth pair (x_{19}, x_{20}) = (7,3) according to the grey scale value for the tenth pair set into 0.65 (grey).

- Filling cell (7,3) for the eleven pair (x_{21}, x_{22}) = (7,3) according to the grey scale value for the eleven pair set into 0.67 (light grey).

- Filling cell (6,2) for the twelve pair (x_{23}, x_{24}) = (6,2) according to the grey scale value for the twelve pair set into 0.69 (light grey).

- Filling cell (7,2) for the thirteen pair (x_{25}, x_{26}) = (7,2) according to the grey scale value for the thirteen pair set into 0.72 (light grey).

- Filling cell (7,2) for the fourteen pair (x_{27}, x_{28}) = (7,2) according to the grey scale value for the fourteen pair set into 0.75 (very light grey).

- Filling cell (6,3) for the fifteen pair (x_{29}, x_{30}) = (6,3) according to the grey scale value for the fifteen pair set into 0.77 (very light grey).

- Filling cell (7,2) for the sixteen pair (x_{31}, x_{32}) = (7,2) according to the grey scale value for the sixteen pair set into 0.79 (very light grey).

- Filling cell (5,2) for the seventeen pair (x_{33}, x_{34}) = (5,2) according to the grey scale value for the seventeen pair set into 0.85 (very light grey).

The order of intensities allows full restoration of the order of the pairs and their values from the image if all pairs differ. If some pairs are equal, they will collide in the image location. In this case, the intensity of the last pair is shown in the image in this basic version. If each cell uses only a single pixel in step 2 in CPC-R version 1.0, a small image

with 10×10 pixels will represent a 17-D point, when each attribute has 34 different values, by locating ten grey scale pixels in this image.

Fig. 3 is one of the CPC-R images produced from the Ionosphere data and it shows the result after implementing the CPC-R algorithm in Python. Fig. 2 describes the process of generation of Ionosphere in Excel.

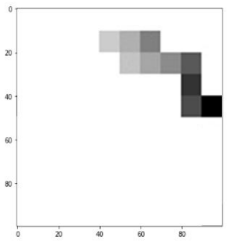


FIGURE 3: Example image of CPC-R

This visualization is lossless if values of all pairs (x_i, x_{i+1}) do not repeat. Below we present the treatment of such colliding pairs and other steps in more elaborated versions of the CPC-R algorithm.

CPC-R version 2.1 With Adjacent Cells

1. Set image coordinate system (origin at upper left, or low left).
2. Locate non-colliding pairs (x_i, x_{i+1}) in the image at cell coordinate values (x_i, x_{i+1}).
3. Set the starting adjacent cell (right, left, top or bottom) for collided pairs.
4. Set order of filling of adjacent cells (clockwise or counterclockwise).
5. Locate colliding pairs (x_i, x_{i+1}) in the cell adjacent to the cell (x_i, x_{i+1}) according to steps 3 and 4.

CPC-R version 2.2 With Splitting Cells

1. Set image coordinate system (origin upper left, or low left).
2. Locate non-colliding pairs (x_i, x_{i+1}) in the image at cell coordinate values (x_i, x_{i+1}).
3. Split colliding cells to vertical strips.
4. Assign the grey scale intensity from black for (x_1, x_2) and very light grey for (x_{n-1}, x_n) for non-colliding pairs. Alternatively, assign heatmap colors for non-colliding pairs.
5. Assign the grey level intensity of the respective pair of values (x_i, x_{i+1}) to the strip for colliding pairs.

   For instance, a cell 8 $\times$ 8 will consist of two strips 4$\times$8 when two pairs are collided. For three colliding pairs, three strips 3$\times$8, 3$\times$8, and 2$\times$8 are produced. For four colliding pairs, four strips 2$\times$8, are produces and for five pairs, five strips 2x8, 2$\times$8, 2$\times$8, 1$\times$8, and 1$\times$8 are produced.

CPC-R version 2.3 With the Darkest Intensity Of Colliding Pairs

1. Assign the grey scale intensity from black for (x_1, x_2) and very light grey for (x_{n-1}, x_n) for non-colliding pairs. Alternatively, assign heatmap colors for non-colliding pairs.
2. Assign the darkest grey level intensity of all pairs with equal values (x_i, x_{i+1}) for colliding pairs.
3. Put an image of each n-D point on the top of mean images of the training cases and the images are put side by side, to form a double image.

CPC-R version 3.0

The CPC-R version 3.0 with incorporated optimized intensities and pairing of coordinates (x_i, x_j).

1. Optimize intensities and pairing coordinates beyond sequential pairs (x_i, x_{i+1}) by methods ranged from random generating a fixed number of alternatives and testing ML prediction accuracy on training data for these alternatives to genetic algorithms.

This approach *enriches the images with context* for analysis by humans and deep learning algorithms. The mean images visualize the difference between two means in its 34-D richness without degrading it to a lossy single distance number. In all experiments below with data from the UCI ML repository, the 10-fold Cross Validation (CV) is used in all experiments.

CHAPTER III

EXPERIMENTS WITH DATA SETS

The purpose of performing all experiments is to improve the images for human visual analysis and improve the deep learning CNN algorithms results. Multiple experiments have been conducted to explore the impact of combinations of properties of images on the accuracy of classification.

The Ionosphere, Glass, and Car data are taken from the UCI Machine Learning repository [8]. The results reported below are obtained in the computational experiments with the Ionosphere data (351 instances of 2 classes with 34 attributes) [7], the Glass data (214 instances of 6 classes with 10 attributes) [9], and the Car data (1728 instances of 4 classes with 6 attributes) [10]. These experiments with Ionosphere, Glass and Car data sets are to test the feasibility of the CPC-R approach for CNN classifiers, and the different number of pixels per cell, which represents each pair (x_i, x_{i+1}).

The grey scale intensities are normalized from 0 to 1 for these experiments and are assigned as follow for Ionosphere data: $I(x_1, x_2) = 0$ (black), $I(x_3, x_4) = 0.2$ (very dark grey), $I(x_5, x_6) = 0.3$ (very dark grey), $I(x_7, x_8) = 0.35$ (very dark grey), $I(x_9, x_{10}) = 0.4$ (dark grey), $I(x_{11}, x_{12}) = 0.45$ (dark grey), $I(x_{13}, x_{14}) = 0.5$ (grey), $I(x_{15}, x_{16}) = 0.55$ (grey), $I(x_{17}, x_{18}) = 0.60$ (grey), $I(x_{19}, x_{20}) = 0.65$ (grey), $I(x_{21}, x_{22}) = 0.67$ (light grey), $I(x_{23}, x_{24}) = 0.69$ (light grey), $I(x_{25}, x_{26}) = 0.72$ (light grey), $I(x_{27}, x_{28}) = 0.75$ (light grey), $I(x_{29}, x_{30}) = 0.77$ (very light grey), $I(x_{31}, x_{32}) = 0.79$ (very light grey), $I(x_{33}, x_{34}) = 0.85$ (very light grey).

The intensities and pairing coordinates (x_i, x_j) are optimized for some of these experiments to obtain better accuracy.

The experiments are done in three ways:

- Initial experiments (CPC-R version 1.0).
- Experiments dealing with colliding values (CPC-R version 2.1).
- Experiments with double images and padding (CPC-R version 2.1, 2.2, 2.3, and 3.0).

Initial Experiments (E1)

Initial Experiments with CNN classifiers

This experiment E1 with CPC-R version 1.0 and grey scale defined. Used two different types of splitting methods to split the data. First is the randomly selected *Train-Test Split* method and next is the randomly selected *10-fold Cross Validation* method. The split percentage for *Train-Test Split* is 90 and 10, which is 90% of train data and 10% of test data.

Initial work with the CNN classifier used in [3] that we denote as CNN64 for short and summarize below to be able to compare results:

- Convolutional Layer with 64 output channels, a kernel shape of 2x2, the stride of 2x2, and RELU activation.
- Convolutional Layer with 64 output channels, a kernel shape of 2x2, the stride of 2x2, and RELU activation.
- Pooling Layer with pooling size of 2x2.
- Drop out Layer with a fraction of input units to drop set to 0.4.
- Convolutional Layer with 128 output channels, a kernel shape of 2x2, the stride of 2x2, and RELU activation RELU.

- Convolutional Layer with 128 output channels, a kernel shape of 2x2, the stride of 2x2, and relu activation.
- Pooling Layer with pooling size of 2x2.
- Drop out Layer with a fraction of input units to drop set to 0.40.
- Fully connected Layer with 256 output nodes and RELU activation.
- Drop out Layer with a fraction of input units to drop 0.40.
- Fully connected Layer with the number of output nodes equal to the number of classes, with a softmax activation.

Table 1 shows the results for this E1 experiment with CNN64. When I used the Convolutional Layers with 64 and 128 output channels, noticed the accuracy is a little lower compared to the Table 1 results. Next, added another each two layers. The above mentioned CNN64 classifier is used for all experiments to get better accuracy. The best accuracy result for the *Train-Test Split* procedure is 79.85% with 150 epochs and the image size is 100×100 for Ionosphere data. The best accuracy result for the *Train-Test Split* procedure is 91.63% with 100 epochs and the image size is 50×50 for Glass data.

TABLE 1: Results for this E1 experiment with CNN64 classifier.

Data	Epochs	Image Size	Cross validation	Accuracy
Ionosphere	250	100×100	10-fold	84.3
Ionosphere	100	50×50	Stratified 10-fold	76.54
Glass	100	100×100	10-fold	96.45
Glass	60	50×50	Stratified 10-fold	94.65

In all experiments, we explore the impact of image sizes and origins of coordinates in the low left corner (LLC).

The next experiment was conducted with another CNN classifier - Inception ResNetV2 classifier to explore if it will produce a better result.

$$Flatten \rightarrow Dense(256, relu) \rightarrow Dense(1, sigmoid).$$

Table 2 shows the results for this E1 experiment with the Inception ResNetV2 classifier. It provided a slightly better accuracy for Ionosphere data (86.95% vs. 84.3%) but somewhat worse accuracy for Glass data (95.45% vs. 96.45%) in comparison with the initial experiment (see Tables 1-2).

TABLE 2: Results for this E1 experiment with Inception ResNetV2 classifier.

Data	Epochs	Image Size	Cross Validation	Accuracy
Ionosphere	250	100×100	10-fold	86.98
Ionosphere	100	50×50	Stratified 10-fold	85.98
Glass	250	100×100	10-fold	95.45
Glass	100	50×50	Stratified 10-fold	95.58

Initial Experiments with MLP classifier

To check how important is the use of DL models vs. simpler MLP models, we conducted experiments with the following Multilayer Perceptron (MLP) classifier that is the simplification of CNN64:

- Layer with 64 output channels and RELU activation. This hidden layer with 64 nodes and a rectified linear activation function.
- Layer with 64 output channels and RELU activation. This hidden layer with 64 nodes and a rectified linear activation function.
- Drop out layer with a fraction of input units to drop set to 0.4.
- Layer with 128 output channels and RELU activation.

– Drop out layer with a fraction of input units to drop set to 0.4.

– Fully connected layer with the number of output nodes equal to the number of classes, with a ‘softmax’ activation.

Table 3 shows the results for both Ionosphere data and Glass data with the MLP classifier. When I used the Layers with 64 and 128 output channels, noticed the accuracy is a little lower compared to the Table 3 results. Next, added other layer with another 64 output channel for better results. The above mentioned MLP classifier is used for all experiments to get better accuracy.

TABLE 3: Results for this E1 experiment with MLP classifier.

Data	Epochs	Image Size	Cross validation	Accuracy
Ionosphere	100	50×50	10-fold	72.5
Ionosphere	100	50×50	Stratified 10-fold	74.58
Glass	100	100×100	10-fold	89.8
Glass	60	100×100	Stratified 10-fold	86.14

Difference between the model CNN64 and MLP

The Convolution layers with 64 and 128 output channels and filters (each of which uses a window size) are added in the CNN classifier. The ”MaxPooling” layer is placed after each convolution layer with the same window size. The sequential model is first defined in the CNN classifier. This combination is applied several times. The width and height of the image get reduced. But the dimensions are added to the image.

The sequential model is first defined in MLP and added a flatten layer to convert it into a flattened vector. Those are the significant differences between CNN and MLP. MLP used a simple data vector. The fully connected ”Dense” layer with 64 neurons is added in MLP. This combination is applied several times with different output channels [11].

The observed difference between CNN and MLP classifiers is as follows:

– CNN has layers of ‘convolution and pooling’. Convolutional layers take advantage of the local spatial coherence of the input. The only possible input correlations are assumed to be spatially close. MLP ignores spatial information. The inputs for the MLP is flattened vectors.

– In this model, CNN can cut down on the number of parameters by sharing weights, and in image processing, it is more efficient compared to multi-layer perceptron. In MLP, the number of total parameters can increases to very high (number of neurons in layer 1 multiplied by of m in layer 2 multiplied by of m in layer 3. . .). There is some redundancy in high dimensions, and it is inefficient.

– The pooling layers are used in CNN. The inputs are reduced more than the number of parameters. In the image classification, this is important, where the last layers of the network need to determine whether an object is present in the scene, but not where [12].

Experiments Dealing With Colliding Values (E2)

This E2 experiment is with the CPC-R version 2.1. Some pairs of values can be equal and collide with each other. The initial experiments above dealt with putting the last pair’s intensity as a cell intensity in CPC-R. It means that other colliding pairs are missed. This experiment E2 is performed with grey CPC-R images, with two different ways (orders) to fill adjacent cells [13] [14], and two Deep Learning CNN classifiers. Fig. 4 illustrates adjacent filling cells for one of the cells in the middle, and Table 4 and 5 show the results.

FIGURE 4: Adjacent cells.

Below are the two used orders of filling adjacent cells:

- $right \rightarrow down \rightarrow left \rightarrow up$
- $left \rightarrow up \rightarrow right \rightarrow down$

Tables 4 and 5 reported the results for replacing the colliding values into adjacent values with CNN, Inception ResNetV2, and MLP classifiers for the Ionosphere data and Glass data. The data is trained and tested with a Image Classification ML model and Keras backed by Tensor-Flow in python.

TABLE 4: Results for Colliding values experiment E2 with Ionosphere data.

Model	Epochs	Image Size	Cross validation	Accuracy
CNN64	100	30×30	10-fold	89.04
CNN64	60	100×100	Stratified 10-fold	85.41
ResNetV2	100	100×100	10-fold	85.8
ResNetV2	100	50×50	Stratified 10-fold	89.98
MLP	100	50×50	10-fold	87.41
MLP	60	50×50	Stratified 10-fold	87.43

TABLE 5: Results for Colliding values experiment E2 with Glass data.

Model	Epochs	Image Size	Cross validation	Accuracy
CNN64	100	100×100	10-fold	95.71
CNN64	60	100×100	Stratified 10-fold	96.1
ResNetV2	200	100×100	10-fold	96.86
ResNetV2	100	50×50	Stratified 10-fold	95.87
MLP	60	50×50	Stratified 10-fold	94.6

This experiment E2 provided a slightly better accuracy for Ionosphere data (89.98% vs. 86.95%) but somewhat worse accuracy for Glass data (96.86% vs. 95.58%) in comparison with the best prior results described (see Tables 1-5). The best accuracy in Table 4 is 89.98% with the ResNetV2 classifier, which is a better accuracy compared to previous experiment E1 (86.95%). The best accuracy in Table 5 is 96.86% with the ResNetV2 classifier, which is a better accuracy compared to previous experiment E1 (95.58%).

Experiments With Double Images And Padding

Experiments E3-E7 with CPC-R version 2.1, 2.2, 2.3, and 3.0 explore the impact of adding context via double images on the accuracy for different classifiers, image sizes, and other parameters listed below for each E3-E7. It is testing a hypothesis that adding the context can increase accuracy. The reverse of these experiments is exploring the impact of these combinations of properties of images on the accuracy of classification.

Experiment E3 with double images and padding

This experiment E3 is with the CPC-R version 2.1, 2.2, 2.3, and 3.0. In this experiment E3, each GLC-R image is superposed on both classes for Ionosphere data G mean (G mean image of G class in Ionosphere data) and B mean (B mean image of B Class in Ionosphere data).

- Gray Image is formed in CPC, and colliding cells are handled by splitting cells. The red image is superposed on both class's mean (gray).
- Both the superposed images are put side by side to form a double image.

– After forming a double image, the image's dimension is $(w) \times (2 * w)$. Then $(2 * w) \times (2 * w)$ image is formed by padding the double image.

Figs. 5-8 illustrate the process of constructing a double image and superimposed with mean images for each class from training data with padding.

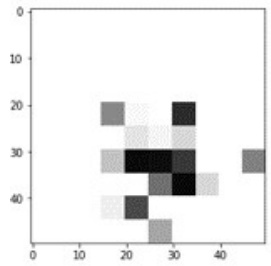


FIGURE 5: Image formed after marking adjacent cells to handle collisions.

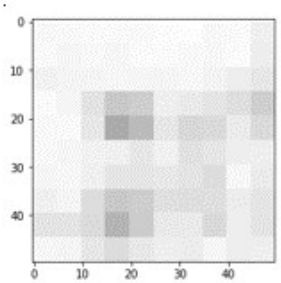


FIGURE 6: G mean image – mean of all images in the training set of a particular fold with a G label.

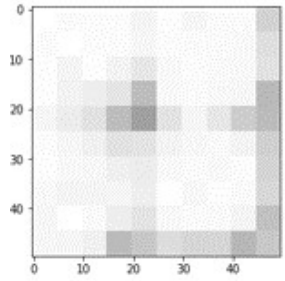


FIGURE 7: B mean image – mean of all images in the training set of a particular fold with a B label.

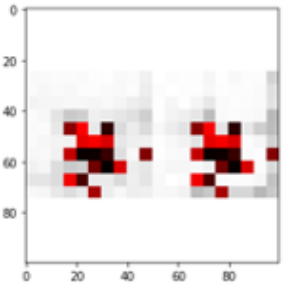


FIGURE 8: Final double image- the image with red levels superimposed with mean images.

The goal for performing this experiment is to improve the images for human visual analysis and improve the deep learning CNN algorithms results.

The GLC-R images (see Figs. 5-8) are generated in this experiment E3, with red levels and gray means, with optimized intensities and pairing of coordinates (x_i, x_j).

Tables 6 and 7 show the results for this E3 experiment and it proves that the optimization of pairing was most beneficial to improve the accuracy, while the accuracy can vary due to the DL random elements. One of the results of this experiment E3 is slightly better than in the experiments presented in E1-E2.

The optimization method used was a simple one: random generating a fixed number of perturbed alternatives and testing ML prediction accuracy on training data for these alternatives. It indicates that more sophisticated methods can improve results further.

TABLE 6: Results for gray level and red means (experiment E3 with Ionosphere data).

Epochs	Image Size	Model	Cross Validation	Accuracy
60	50×50	CNN64	10-fold	89.46
100	100×100	CNN64	10-fold	91.47
100	50×50	CNN64	Stratified 10-fold	89.13
100	100×100	ResNetV2	10-fold	92.03
100	100×100	ResNetV2	Stratified 10-fold	91.30
100	50×50	MLP	10-fold	88.46
60	50×50	MLP	Stratified 10-fold	87.19

TABLE 7: Results for gray level and red means (experiment E3 with Ionosphere data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	50×50	CNN64	10-fold	93.80
100	50×50	CNN64	Stratified 10-fold	94.13
100	50×50	ResNetV2	10-fold	93.80
100	100×100	ResNetV2	Stratified 10-fold	92.03
200	100×100	MLP	10-fold	95.8
60	100×100	MLP	Stratified 10-fold	95.89

Experiment E3 provided a better accuracy for Ionosphere data (92.03% vs. 89.98%), but worse accuracy for Glass data (95.89% vs. 96.86%) in comparison with the best results of experiments E1 and E2 (see Tables 1-5). The best accuracy in Table 6 is 92.03% with the ResNetV2 classifier, which is a better accuracy compared to previous

experiment E2 (89.98%). The best accuracy in Table 7 is 95.89% with the MLP classifier, which is a little worse accuracy compared to previous experiment E1 (96.86%).

In this experiment E3, the MLP provided the better accuracy for Glass data which is 95.89% with *Stratified 10-fold cross validation* and *large image size* 100×100. There is not much difference between the accuracies after increasing the epochs in Table 7. The MLP classifier is a competitive classifier to other CNN classifiers for both Ionosphere data and Glass data.

Experiment E4 with double images and padding

- This experiment E4 is with the CPC-R version 2.1, 2.2, 2.3, and 3.0. The GLC-R images are generated with several gray level cases and colored double image combinations are formed, and colliding cells are resolved by marking the adjacent cells.

- A colored image is formed as described below, and colliding cells are handled by marking adjacent cells.

  Then the image is converted into a gray image and is superposed on colored means of classes.

- Each color cell's intensity is chosen randomly (3 intensities for 3 channels of a cell).

- The Ionosphere data are generated with 51 random values for 17 pairs and 15 random values for 5 pairs are generated for the Glass data.

- Figs. 9-12 illustrate the process of constructing a double image and superimposed with mean images for each class from training data with padding.

– The GLC-R images (see Figs. 9-12) are generated in experiment E4, with gray levels and colored means, with optimized intensities and pairing of coordinates (x_i, x_j).

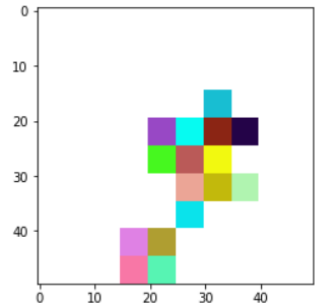


FIGURE 9: Image formed after marking adjacent cells to handle collisions.

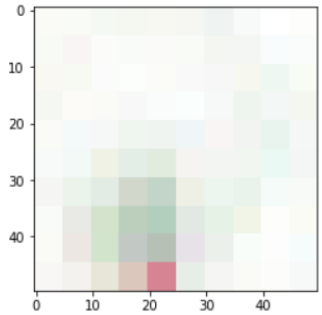


FIGURE 10: G mean image – mean of all images in the training set of a particular fold with a G label.

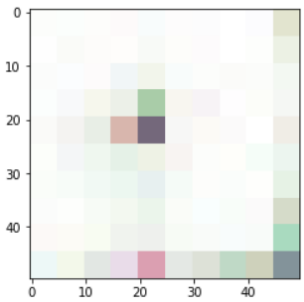


FIGURE 11: B mean image - mean of all images in the training set of a particular fold with a B label.

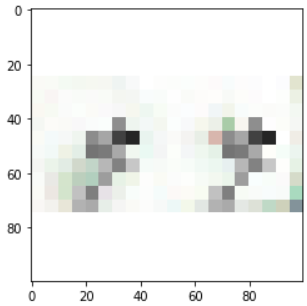


FIGURE 12: Final double image – the image with gray levels superimposed with mean images.

Tables 8 and 9 show the results of the E4 experiment with Ionosphere data and Glass data.

TABLE 8: Results for gray level and colored means (experiment E4 with Ionosphere data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	50×50	CNN64	10-fold	91.73
60	50×50	CNN64	Stratified 10-Fold	91.71
100	50×50	ResNetV2	10-fold	91.80
60	50×50	ResNetV2	Stratified 10-Fold	91.25
100	100×100	MLP	10-fold	90.91

TABLE 9: Results for gray level and colored means (experiment E4 with Glass data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	50×50	CNN64	10-fold	93.80
100	100×100	CNN64	10-fold	95.9
100	50×50	CNN64	Stratified 10-Fold	93.8
100	50×50	ResNetV2	10-fold	92.7
60	50×50	ResNetV2	10-fold	90.05
100	50×50	ResNetV2	Stratified 10-Fold	91.87
100	50×50	MLP	10-fold	94.70
60	50×50	MLP	Stratified 10-fold	93.07

Experiment E4 provided a slightly worse accuracy for both Ionosphere data (91.80% vs. 92.03%) and Glass data (95.9% vs. 96.45%) in comparison with the best results of experiments E1-E3 (see Tables 1-7). The best accuracy in Table 8 is 91.80% with the ResNetV2 classifier, which is a little worse accuracy compared to previous experiment E3 (92.03%). The best accuracy in Table 9 is 95.9% with CNN64 classifier, which is a little worse accuracy compared to previous experiment E1 (96.86%).

Table 8, the MLP provided a little worse accuracy than the other CNN classifiers for Ionosphere data. There is much difference between the accuracies. In the comparison of CNN and MLP in Table 9, MLP provided better accuracy than CNN with the same image size and Cross Validation. The MLP is a competitive classifier to other CNN classifiers.

Experiment E5 with double images and padding

- This experiment E5 is with the CPC-R version 2.1, 2.2, 2.3, and 3.0. The GLC-R images are generated with cases at red levels and colored means with results presented in Tables 10 and 11.
- Image (colored) is formed in CPC, and colliding cells are handled by marking adjacent cells. Then the image (converted in red) is superposed on classes mean (colored).
- Both the superposed images are put side by side to form a double image. After forming a double image, the dimension of the image is (w) x (2*w). Then a ((2*w) x (2*w)) image is formed by padding the double image.
- Similarly to the previous experiments (E3-E4), each cell's intensity is chosen randomly (3 intensity for 3 channels of a cell). The Ionosphere data are generated with 51 random values for 17 pairs and 15 random values for 5 pairs are generated for the Glass data.

The GLC-R images (see Figs. 13-16) are generated in this experiment E5, with red levels and colored means, with optimized intensities and pairing of coordinates (x_i, x_j).

The results for this experiment E5 summarized in Tables 10 and 11:

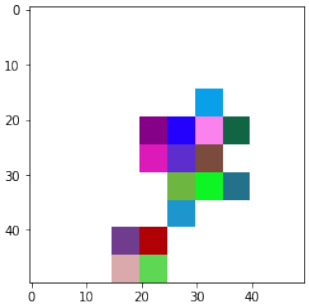


FIGURE 13: Image formed after marking adjacent cells to handle collisions.

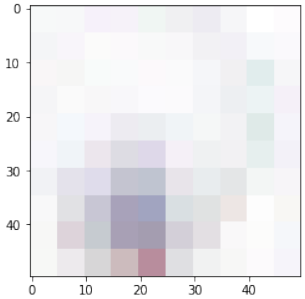


FIGURE 14: G mean image–mean of all images in the training set of a particular fold with a G label.

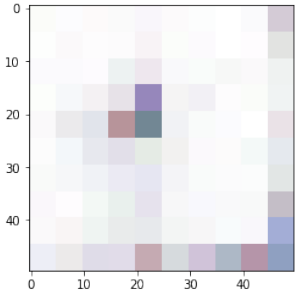


FIGURE 15: B mean image–mean of all images in the training set of a particular fold with a B label.

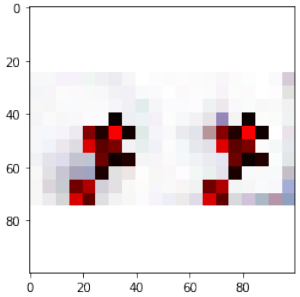


FIGURE 16: Final double image – the image with red levels superimposed with mean images.

TABLE 10: Results for red level and colored means (experiment E5 with Ionosphere data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	100×100	CNN64	10-fold	93.46
100	50×50	CNN64	Stratified 10-Fold	93.18
100	50×50	ResNetV2	10-fold	91.80
100	50×50	ResNetV2	Stratified 10-Fold	91.80
100	100×100	MLP	10-fold	90.91
60	100×100	MLP	Stratified 10-fold	85.75

TABLE 11: Results for red level and colored means (experiment E5 with Glass data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	50×50	CNN64	10-fold	94.85
100	50×50	CNN64	Stratified 10-Fold	94.05
100	50×50	ResNetV2	10-fold	95.06
100	50×50	ResNetV2	Stratified 10-Fold	94.89
60	100×100	MLP	Stratified 10-fold	92.54

Experiment E5 provided a slightly better accuracy for both Ionosphere data (93.46% vs. 92.03%) but a little worse for Glass data (95.06% vs. 96.86%) in comparison with the best results of experiments E1-E4 (see Tables 1-9). The best accuracy in Table 10 is 93.46% with the CNN64 classifier, which is a better accuracy compared to previous experiment E3 (92.03%). The best accuracy in Table 11 is 95.06% with the CNN64 classifier, which is a little worse accuracy compared to previous experiment E1 (96.86%).

Table 10 and 11, the MLP provided the worse accuracy than the other CNN classifiers for both Ionosphere data and Glass data. There is much difference between the accuracies. In the comparison of classifiers in Table 11, there is not much difference between the accuracies with the same image size and Cross Validation.

Experiment E6 with double images and padding

This experiment E6 is with the CPC-R version 2.1, 2.2, 2.3, and 3.0. The GLC-R images are generated with cases at colored levels and colored means (see Figs. 17-20).

- A color image is formed in CPC-R, and colliding cells are handled by marking adjacent cells. Then the image is superposed on colored mean images of classes.

- Both superposed images are put side by side to form a double image. After forming a double image, the dimension of the image is (w) x (2*w).

- Then a ((2*w) x (2*w)) image is formed by padding the double image.

– Each cell intensity is chosen randomly (3 intensity for 3 channels of a cell). The Ionosphere data are generated with 51 random values for 17 pairs and 15 random values for 5 pairs are generated for the Glass data.

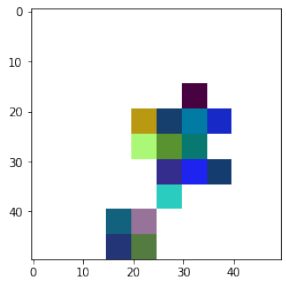


FIGURE 17: Image formed after marking adjacent cells to handle collisions.

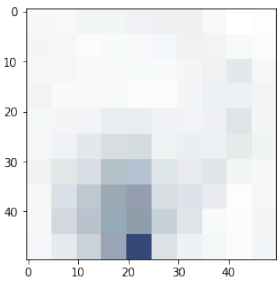


FIGURE 18: G mean image – mean of all images in the training set of a particular fold with a G label.

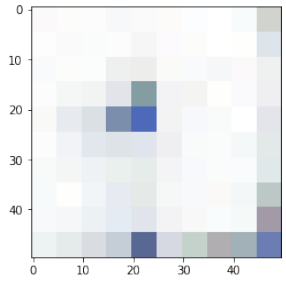


FIGURE 19: B mean image–mean of all images in the training set of a particular fold with a B label.

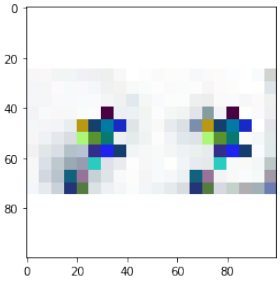


FIGURE 20: Final double image – the image with colored levels superimposed with mean
images.

The GLC-R images (see Figs. 17-20) are generated in experiment E6, with red levels and colored means, with optimized intensities and pairing of coordinates (x_i, x_j). Tables 12 and 13 show the results for Ionosphere data and Glass data.

TABLE 12: Results for colored levels and means (experiment E6 with Ionosphere data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	100×100	CNN64	10-fold	92.46
100	50×50	CNN64	Stratified 10-Fold	90.86
100	50×50	ResNetV2	10-fold	91.80
100	50×50	ResNetV2	Stratified 10-fold	90.04
60	50×50	MLP	10-fold	90.29
60	100×100	MLP	Stratified 10-fold	86.51

TABLE 13: Results for colored levels and means (experiment E6 with Glass data).

100	100×100	CNN64	10-fold	95.68
100	50×50	CNN64	Stratified 10-Fold	94.86
100	50×50	ResNetV2	10-fold	96.80
100	50×50	ResNetV2	Stratified 10-fold	94.23
100	100×100	MLP	10-fold	93.28
60	100×100	MLP	Stratified 10-fold	91.28

Experiment E6 provided a little worse accuracy for both Ionosphere data (92.46% vs. 93.46%) but a slightly better accuracy for Glass data (96.45% vs. 96.80%) in comparison with the best results of experiments E1-E5 (see Tables 1-11).

The best accuracy in Table 12 is 92.46%, which is a worse accuracy compared to previous experiments E5 (93.46%). The best accuracy in Table 13 is 96.80%, which is a slightly worse accuracy compared to previous experiments E1 (96.86%). MLP classifier provided worse accuracy than other CNN classifiers in Tables 12 and 13.

Experiment E7 with red on grey and confusion matrices

This experiment E7 is with the CPC-R version 2.1, 2.2, 2.3, and 3.0. Each GLC-R image is superposed on both classes for Ionosphere data G mean (G mean image of G class in Ionosphere data) and B mean (B mean image of B class in Ionosphere data). This experiment E7 generated the GLC-R images (see Figs. 22-25), with red levels and gray means, with optimized intensities and pairing of coordinates (x_i, x_j).

– The gray image is formed in CPC, and colliding cells are handled by splitting cells. A red image is superposed on the grey means of both classes. Both the superposed images are put side by side to form a double image.

– After forming a double image, the dimension of the image is (w) x (2*w). Then (2*w) x (2*w) image is formed by padding the double image.

– The intensity of each point is chosen randomly (one intensity value for a point). The Ionosphere data contains 51 random values for 17 pairs.

– Each case from training and validation data is labeled as BB, GG, GB, BG for Ionosphere data to reflect the prediction by a CNN model, e.g., gg stands for the situation when a case from class g was predicted as g by the model.

Matrices are in the form of Ionosphere data are shown below:

$$\begin{bmatrix} PredictedGActualG & PredictedGActualB \\ PredictedBActualG & PredictedBActualB \end{bmatrix}$$

```
length of X_train: 315
length of X_val: 36
train_confusion matrix ::
 [[ 91  17]
 [ 23 184]]
val_confusion matrix ::
 [[15  3]
 [ 1 17]]
```

FIGURE 21: Confusion matrices for Ionosphere data.

Figs. 22-25 visualize a case in red levels, mean images of two classes, and a case superimposed with mean images placed side by side from the Ionosphere data. The results are summarized in Table 14 for Ionosphere data and Glass data.

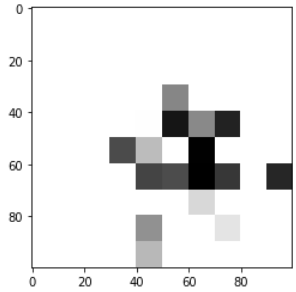


FIGURE 22: Image formed after marking adjacent cells to handle collisions.

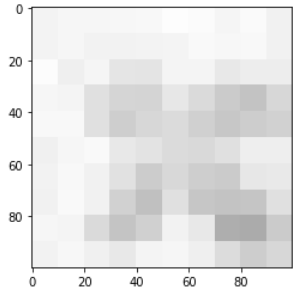


FIGURE 23: G mean image-mean of all images in the training set of a particular fold with a G label.

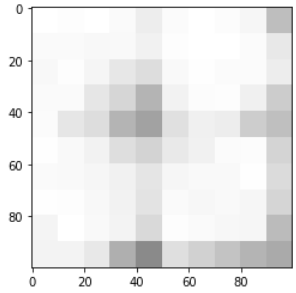


FIGURE 24: B mean image–mean of all images in the training set of a particular fold with a B label.

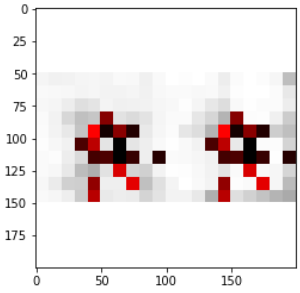


FIGURE 25: Final double image – the image with red levels superimposed with mean
images.

TABLE 14: Results for red levels and gray means with confusion matrices (experiment E7 with Ionosphere data).

Epochs	Image Size	Model	Cross validation	Accuracy
100	50×50	CNN64	10-fold	90.33
100	100×100	CNN64	10-fold	89.96
100	50×50	CNN64	Stratified 10-fold	90.86
100	50×50	ResNetV2	Stratified 10-fold	91.3
100	50×50	MLP	Stratified 10-fold	87.6

Experiment, E7, provided a slightly worse accuracy for Ionosphere data (91.3% vs. 93.46%), in comparison with the best results of experiments E1-E6 (see Table 1-13). The best accuracy in this Table 14 is 91.3%, which is worse accuracy compared to the previous experiment E5 (93.46%).

CHAPTER IV

SALIENCY MAP

The Saliency Map was first witnessed in paper [15]. This Saliency Maps concept plays an important role in Deep learning and Computer vision. The Saliency Map is an image. Saliency Maps brighten the pixel and represents how salient the pixel is. The brightness of the pixel is directly proportional to its saliency. Saliency maps are generally a grey scale image. Saliency maps are also called heat maps. The heat map hotness refers to the areas of the image that have a significant impact on class prediction [16].

The goal of this experiment is to explore abilities to discover the most informative attributes in CPC- R images by using the saliency maps. We explore an approach based on the gradient $\frac{\partial output}{\partial input}$ of the output category with respect to an input image.

It explains how the output value changes the input image pixels with a minor change. Gradients have both positive and negative values, but all positive values show the minor changes in a pixel and increase the output value. This approach expects that visualizing these gradients with the same shape as the image will produce some intuition of attention [17]. These gradients highlight the input regions that can cause a major change in the output and highlight the salient image regions that contribute towards the output [18].

Image-Specific Class Saliency Visualization

The mechanism of this saliency approach follows [15]. Pixels are ranked with a given image I_0, a class c, and a classification ConvNet with the class score function $S_c(I)$, influencing the score $S_c(I_0)$. Example: Consider the linear score model for class c.

$$S_c(I) = w_c^T I + b_c;$$

where image *'I'* = one dimensional form(vectorised).

w_c = weight vector.

b_c = bias of the model.

The magnitude of elements of 'w' explains the importance of the corresponding pixels of an image *I* for the class 'c.'

In deep convolution networks, the class score $S_c(I_0)$ is a very much non-linear function of *I*, so the above line is not applied for this case. Though the given image I_0 can approximate $S_c(I)$ with a linear function in the neighborhood of I_0 by computing the first-order Taylor expansion:

$$S_c(I) \approx \mathrm{w}_c^T I + b_c;$$

where w is the derivative of S_c with respect to the image *I* at the point (image) I_0:

$$\mathrm{w} = \frac{\partial S_c}{\partial I}.$$

The saliency maps are visualized with the highest class score (top1 class prediction) on a randomly selected test image from the data. Such saliency experiments were conducted with CNN on the randomly selected CPC-R test images from Ionosphere data to check the importance of the corresponding pixels in these images.

Saliency Visualisation with Ionosphere data (E8)

There are two techniques used in this experiment.

1. The first technique is generating an image that maximizes the class's score, which visualizing the notion of the class, captured by a Convolutional neural network (CNN) [15][19].

2. The second technique is to compute the class saliency map specific to a given image and class.

This visualization experiment was carried out with a single deep Convolution neural network. This CNN was trained on the Ionosphere data. The initial work was with the CNN classifier used in [3] that we denote as CNN64 for short and summarize below to be able to compare results:

- Convolution layer with 32 output channels, a kernel shape of 3x3, the stride of 3x3, and RELU activation.
- Convolution layer with 64 output channels, a kernel shape of 3x3, the stride of 3x3, and RELU activation.
- Pooling layer with pooling size of 2x2.
- Drop out layer with a fraction of input units to drop set to 0.25.
- Convolution layer with 128 output channels, a kernel shape of 3x3, the stride of 3x3, and RELU activation.
- Convolution layer with 128 output channels, a kernel shape of 3x3, the stride of 3x3, and RELU activation.
- Pooling layer with pooling size of 2x2.
- Drop out layer with a fraction of input units to drop set to 0.50.
- Fully Connected Layer with the number of output nodes equal to the number of classes, with a softmax activation.

First, I used the Convolutional Layers with 64 and 128 output channels, noticed the accuracy is a fraction lower compared to the Table 15 results. Next, added another two Convolutional Layers with 32 and 128 output channels.

In this paper, the CNN models are more transparent by visualizing the regions of input, that are essential for predictions or visual explanation. This approach is called Gradient Weighted Class Activation Mapping, which is (Grad-CAM) [20][21]. It is used in class-specific gradient information flowing into the final Convolutional layer of CNN to generate the localization map of important regions in the image. Grad-CAM is most capable of any CNN classifiers. This experiment also experimented with how the Grad-CAM can combine with Guided BackPropagation to create a high-resolution class discriminative visualization known as Guided Grad- CAM.

Fig. 26-29 shows the representative saliency maps result for CNN64 classifier for one of the CPC-R images from the Ionosphere data: (Fig. 26) Original Image (Fig. 27) Guided BackPropagation, (Fig. 28) Grad-CAM, which localizes class discriminative regions, (Fig. 29) combing of both (Fig. 27) and (Fig. 28), which is Guided Grad- CAM gives the high resolution class discriminative visualization. The gray scale intensities are normalized from 0 to 1.

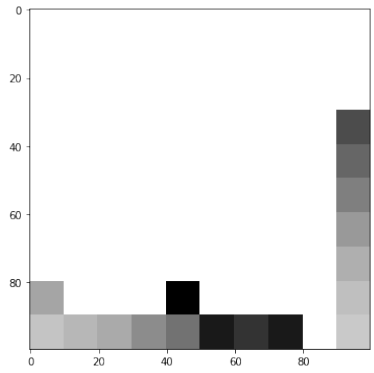


FIGURE 26: Original Image.

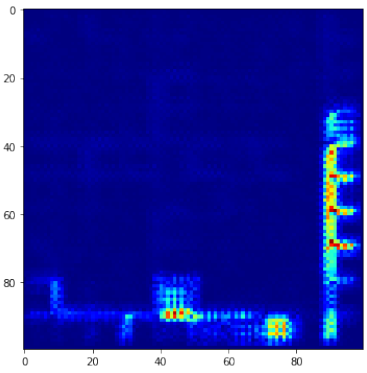


FIGURE 27: Guided BackPropagation.

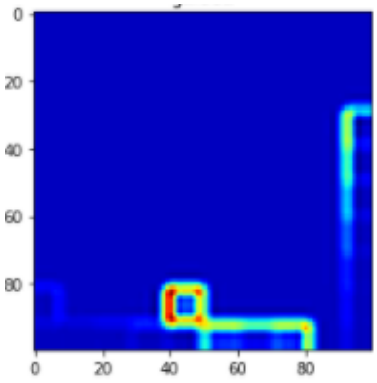


FIGURE 28: Grad-CAM.

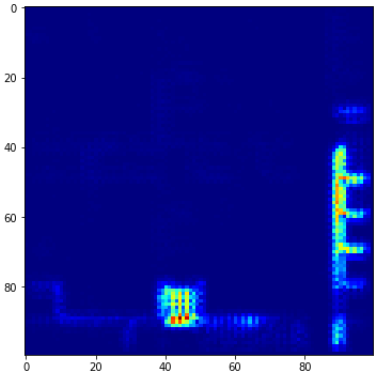


FIGURE 29: Guided Grad-CAM.

Grad-CAM visualizations are the class discriminative images. It will localize the relevant image regions well and doesn't show any finely grained importance, like pixel-space gradient visualization methods. I have chosen Guided BackPropagation over deconvolution because Guided BackPropagation visualizations were generally noise-free [22]. The Grad-CAM can easily localize the darkest pixel in the input region but not others. To combine the best aspects of both, Guided BackPropagation and Grad-CAM visualizations are fused via point-wise multiplication [20]. We can easily locate the dark pixels in Fig. 28 (Grad-CAM). Generated Fig. 28 with Grad-CAM and Fig. 29 Guided Grad-CAM for visual explanation to understand better image classification [23]. This experiment has been conducted using Keras-vis with its components Visualize_saliency and Visual_cam implemented with backpropagation modifiers.

Fig. 30 is from the `Visualize_saliency`. Here the original inputs and their heat maps are showed side by side. These results correspond to the CNN64 network trained on the Ionosphere data. The gray scale intensities are normalized from 0 to 1, the same as the E1-E7 experiments. It will quickly locate the dark pixels in an image.

Fig. 30 contains few CPC-R images from the Ionosphere data with visual_saliency (Guided BackPropagation):

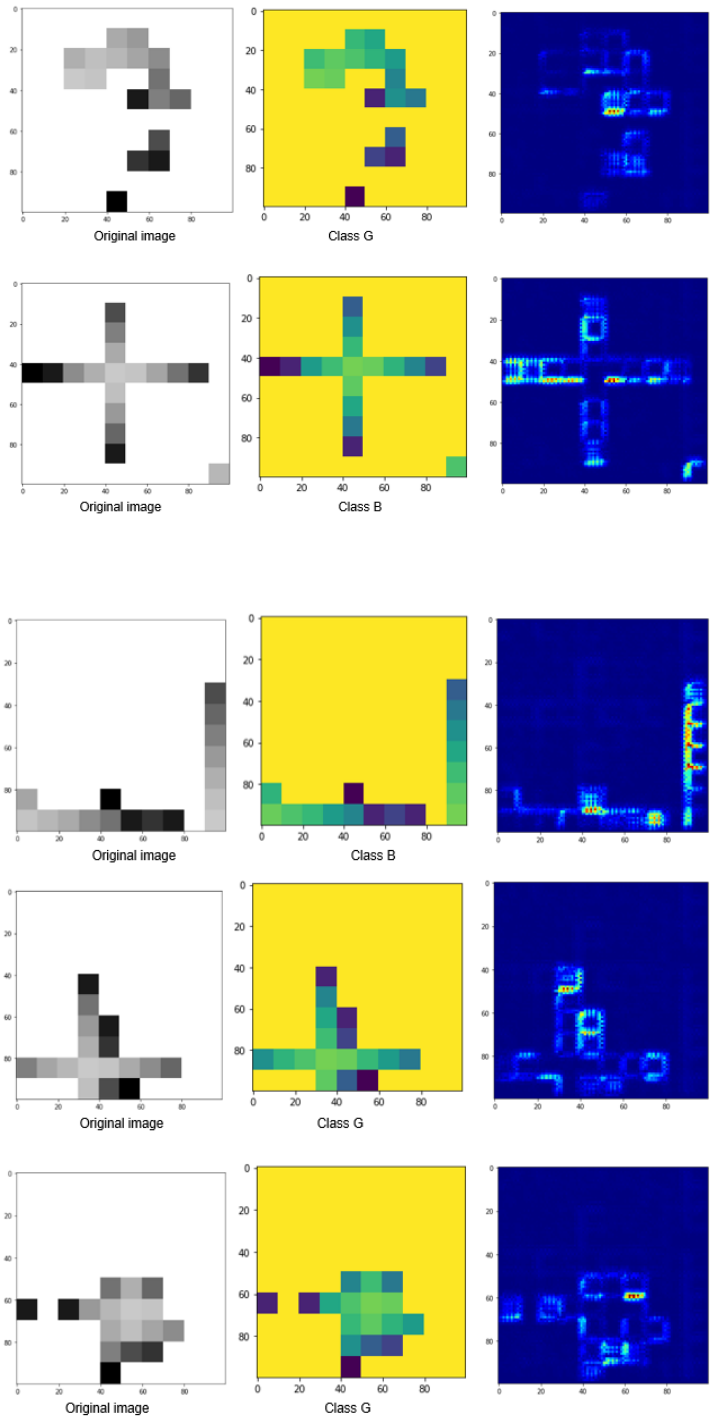


FIGURE 30: Guided BackPropagation.

Fig. 31 contains few CPC-R images from the Ionosphere data with visualize_cam (Grad-CAM):

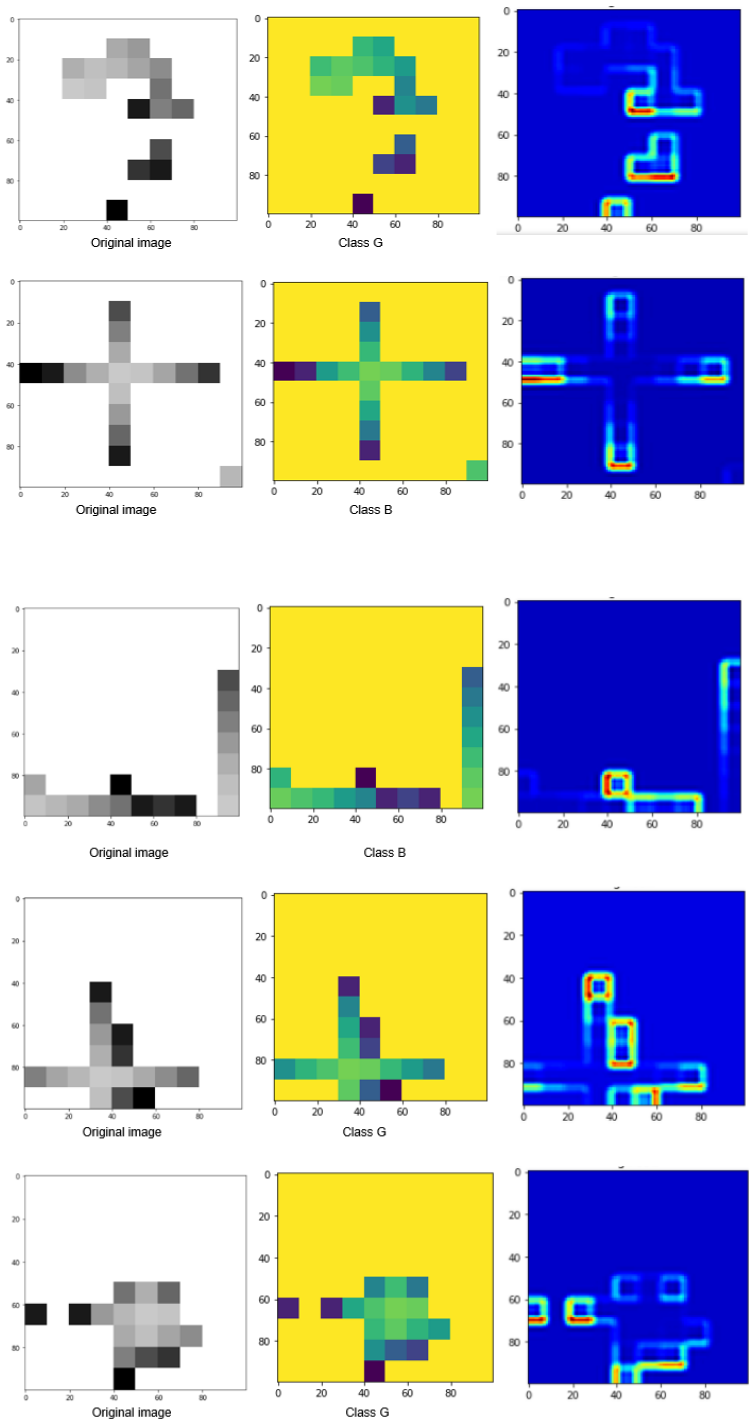


FIGURE 31: Grad-CAM.

Fig. 32 contains few CPC-R images from the Ionosphere data with visualize_cam (Guided Grad-CAM):

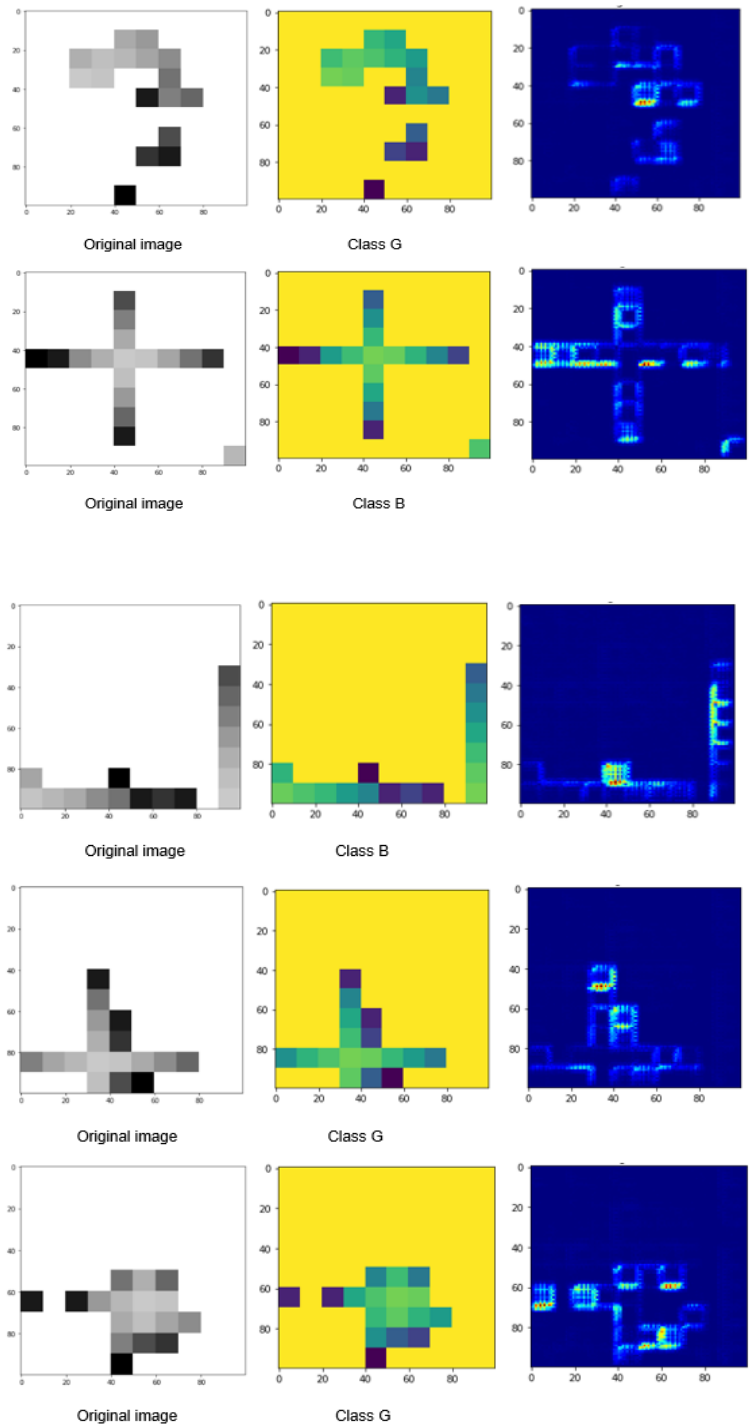


FIGURE 32: Guided Grad-CAM.

In Figs. 30-32, the most salient pixels are visible and correspond to the darkest pixels in CPC-R images. However, the saliency of these pixels is simply an artifact of the CPC-R coding schema where the first pair (x_1, x_2) is the darkest one and the last pair (x_{33}, x_{34}) is the lightest one. Thus, this saliency provides a distorted importance of pixels focusing only on darker start pairs like (x_1, x_2), and so on. The fact that all classifiers provided better accuracy in Table 15 with 10-fold cross validation which says that dark pairs significantly contributed to the output, but it does not justify that they are more important and relevant than others. This is illustrated in the next section where we employ an alternative method to find the importance of the pixels.

TABLE 15: Results for Saliency (experiment E8 with Ionosphere data).

Model	Image Size	Epochs	Cross validation	Accuracy
CNN64	50×50	50	10-fold	94.4
CNN64	50×50	100	Stratified 10-fold	94.58
ResNetV2	100×100	100	Stratified 10-fold	93.46
ResNetV2	50×50	50	10-fold	95.01
MLP	50×50	100	Stratified 10-fold	88.07
MLP	100×100	50	10-fold	90.75

Experiment, E8, provided a better accuracy for Ionosphere data (95.01%vs. 93.46%), in comparison with the best results of experiments E1-E7 (see Table 1-14). The best accuracy in Table 15 is 95.01%, which is better accuracy compared to previous experiment E5 (93.46%).

CHAPTER V

INFORMATIVE CELLS

The purpose of this experiment E9 in the previous section, is to show that saliency maps did not produce the justified interpretable feature importance on CPC-R images focusing only on dark pixels. Another common reason for saliency maps failure is their local approach with small patches without large context, and sensitivity to different contrasts [24]. Thus, we need another method that will be able to deal with these issues better.

The idea of the method presented in this section is to define the importance of features by estimating the change of prediction accuracy due to the exclusion of the individual features. The large cells (super-pixels) are used as such features that can capture a larger context. This approach is in line with the Super CNN approach that is a hierarchical super pixel CNN for salient object detection [24].

This experiment E9 presents the accuracy of classification by covering respective cells of CPC-R images of Ionosphere data that are made white. It is an alternative way to know which cells are most salient/informative. This approach views a cell as the most informative if covering it leads to the largest decrease in classification accuracy. This method is called the Informative Cell Covering (ICC) algorithm.

A 5x5 grid with 25 cells was created for each CPC-R image. Then 25 images were created from each CPC-R image where a respective cell was made white. Table 16 describes the mapping of informative cells.

TABLE 16: Formation of informative cells.

1	Cell 21	Cell 21	Cell 22	Cell 22	Cell 23	Cell 23	Cell 24	Cell 24	Cell 25	Cell 25
2	Cell 21	Cell 21	Cell 22	Cell 22	Cell 23	Cell 23	Cell 24	Cell 24	Cell 25	Cell 25
3	Cell 16	Cell 16	Cell 17	Cell 17	Cell 18	Cell 18	Cell 19	Cell 19	Cell 20	Cell 20
4	Cell 16	Cell 16	Cell 17	Cell 23	Cell 18	Cell 18	Cell 19	Cell 19	Cell 20	Cell 20
5	Cell 11	Cell 11	Cell 12	Cell 12	Cell 13	Cell 13	Cell 14	Cell 14	Cell 15	Cell 15
6	Cell 11	Cell 11	Cell 12	Cell 12	Cell 13	Cell 13	Cell 14	Cell 14	Cell 15	Cell 15
7	Cell 6	Cell 6	Cell 7	Cell 7	Cell 8	Cell 8	Cell 9	Cell 9	Cell 10	Cell 10
8	Cell 6	Cell 6	Cell 7	Cell 7	Cell 8	Cell 8	Cell 9	Cell 9	Cell 10	Cell 10
9	Cell 1	Cell 1	Cell 2	Cell 2	Cell 3	Cell 3	Cell 4	Cell 4	Cell 5	Cell 5
10	Cell 1	Cell 1	Cell 2	Cell 2	Cell 3	Cell 3	Cell 4	Cell 4	Cell 5	Cell 5
0	1	2	3	4	5	6	7	8	9	10

The formation of consecutive pairs of values are shown below:

(x_1, y_{10}), (x_1, y_9), (x_2, y_{10}), (x_2, y_9) = Cell 1;

(x_3, y_{10}), (x_3, y_9), (x_4, y_{10}), (x_4, y_9) = Cell 2;

(x_5, y_{10}), (x_5, y_9), (x_6, y_{10}), (x_6, y_9) = Cell 3;

(x_7, y_{10}), (x_7, y_9), (x_8, y_{10}), (x_8, y_9) = Cell 4;

(x_9, y_{10}), (x_9, y_9), (x_{10}, y_{10}), (x_{10}, y_9) = Cell 5;

(x_1, y_8), (x_1, y_7), (x_2, y_8), (x_2, y_7) = Cell 6;

(x_3, y_8), (x_3, y_7), (x_4, y_8), (x_4, y_7) = Cell 7;

(x_5, y_8), (x_5, y_7), (x_6, y_8), (x_6, y_7) = Cell 8;

(x_7, y_8), (x_7, y_7), (x_8, y_8), (x_8, y_7) = Cell 9;

(x_9, y_8), (x_9, y_7), (x_{10}, y_8), (x_{10}, y_7) = Cell 10;

(x_1, y_6), (x_1, y_5), (x_2, y_6), (x_2, y_5) = Cell 11;

(x_3, y_6), (x_3, y_5), (x_4, y_6), (x_4, y_5) = Cell 12;

(x_5, y_6), (x_5, y_5), (x_6, y_6), (x_6, y_5) = Cell 13;

(x_7, y_6), (x_7, y_5), (x_8, y_6), (x_8, y_5) = Cell14;

(x_9, y_6), (x_9, y_5), (x_{10}, y_6), (x_{10}, y_5) = Cell 15;

(x_1, y_4), (x_1, y_3), (x_2, y_4), (x_2, y_3) = Cell 16;

(x_3, y_4), (x_3, y_3), (x_4, y_4), (x_4, y_3) = Cell 17;

(x_5, y_4), (x_5, y_3), (x_6, y_4), (x_6, y_3) = Cell 18;

(x_7, y_4), (x_7, y_3), (x_8, y_4), (x_8, y_3) = Cell 19;

(x_9, y_4), (x_9, y_3), (x_{10}, y_4), (x_{10}, y_3) = Cell 20;

(x_1, y_2), (x_1, y_1), (x_2, y_2), (x_2, y_1) = Cell 21;

(x_3, y_2), (x_3, y_1), (x_4, y_2), (x_4, y_1) = Cell 22;

(x_5, y_2), (x_5, y_1), (x_6, y_2), (x_6, y_1) = Cell 23;

(x_7, y_2), (x_7, y_1), (x_8, y_2), (x_8, y_1) = Cell 24;

(x_9, y_2), (x_9, y_1), (x_{10}, y_2), (x_{10}, y_1) = Cell 25.

Informative Cell Covering (ICC) with Ionosphere data

Table 17 shows the accuracy of the classification of Ionosphere data with covered cells. These cells are presented in the ascending order of accuracy.

TABLE 17: The classification accuracy of Ionosphere data with covered cells.

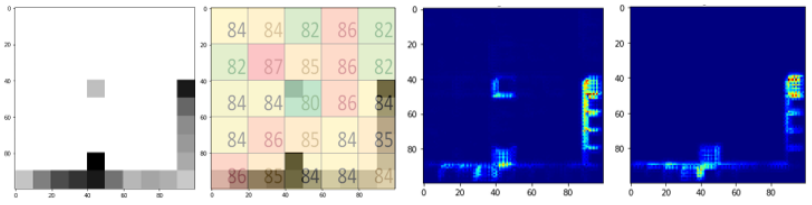
Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
13	100×100	30	CNN64	10-fold	80.35
23	100×100	30	CNN64	10-fold	81.78
25	100×100	30	CNN64	10-fold	82.07
20	100×100	30	CNN64	10-fold	82.34
16	100×100	30	CNN64	10-fold	82.35
11	100×100	30	CNN64	10-fold	83.50

Table 17 (Continued)

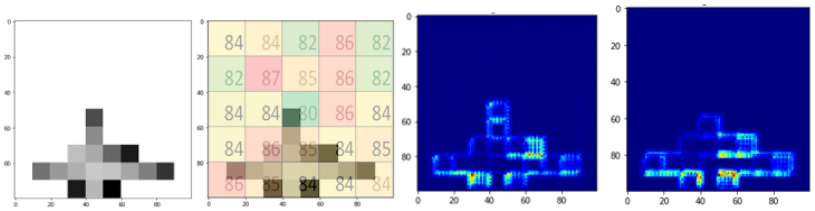
Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
6	100×100	30	CNN64	10-fold	83.77
12	100×100	30	CNN64	10-fold	83.77
3	100×100	30	CNN64	10-fold	83.85
21	100×100	30	CNN64	10-fold	84.05
4	100×100	30	CNN64	10-fold	84.06
9	100×100	30	CNN64	10-fold	84.06
15	100×100	30	CNN64	10-fold	84.06
5	100×100	30	CNN64	10-fold	84.34
22	100×100	30	CNN64	10-fold	84.34
10	100×100	30	CNN64	10-fold	84.61
18	100×100	30	CNN64	10-fold	84.61
2	100×100	30	CNN64	10-fold	84.62
8	100×100	30	CNN64	10-fold	84.64
7	100×100	30	CNN64	10-fold	84.50
14	100×100	30	CNN64	10-fold	85.50
19	100×100	30	CNN64	10-fold	85.50
1	100×100	30	CNN64	10-fold	85.77
24	100×100	30	CNN64	10-fold	85.77
17	100×100	30	CNN64	10-fold	86.62

Comparison of Guided BackPropagation Salient Pixels with ICC Informative Cells

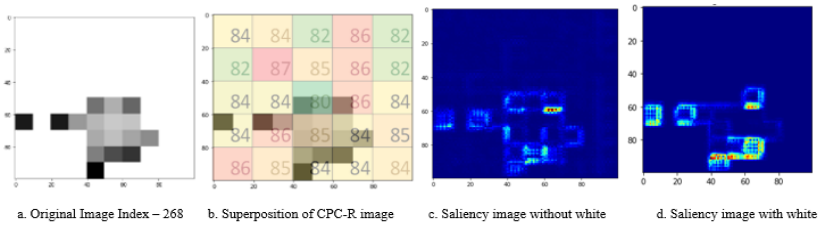
- The lowest accuracy for the ICC method is 80.35% (cell 13), which is considered the most informative in this approach.
- The Guided BackPropagation and ICC results are analyzed in Fig. 33-35 to check saliency and informativeness for cell 13.
- Fig. 33 shows 4 types of images: (a) original CPC-R images from the Ionosphere data, (b) CPC-R images superimposed with heatmap when cell 13 was fully covered, (c) saliency maps for the same CPC-R image without making cell 13 white, and (d) saliency maps for the same CPC-R image with making cell 13 white.



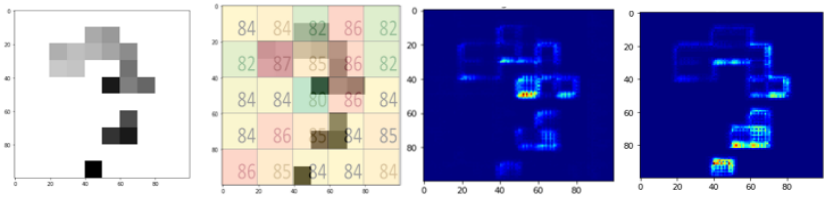
a. Original Image Index – 70 b. Superposition of CPC-R image c. Saliency image without white d. Saliency image with white



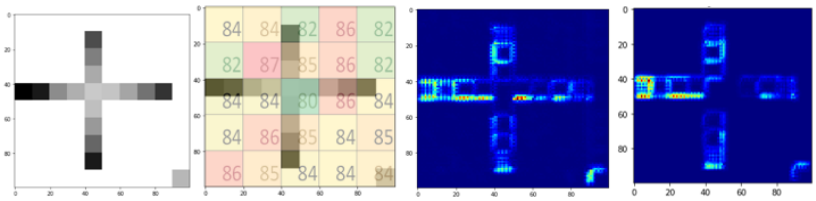
a. Original Image Index – 332 b. Superposition of CPC-R image c. Saliency image without white d. Saliency image with white



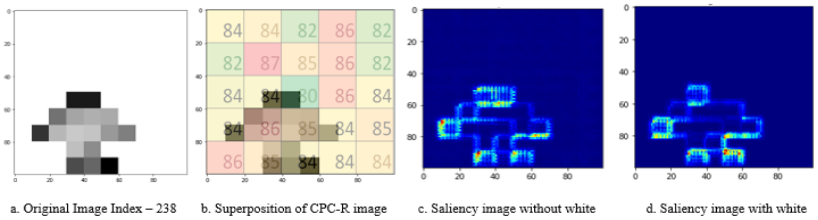
a. Original Image Index – 268 b. Superposition of CPC-R image c. Saliency image without white d. Saliency image with white



a. Original Image Index – 267 b. Superposition of CPC-R image c. Saliency image without white d. Saliency image with white



a. Original Image Index – 249 b. Superposition of CPC-R image c. Saliency image without white d. Saliency image with white



a. Original Image Index – 238 b. Superposition of CPC-R image c. Saliency image without white d. Saliency image with white

FIGURE 33: Informative cell 13.

The second lowest accuracy for the ICC method is 81.78% (cell 23), which is considered the second most informative cell. Fig. 34 contains the same four types of images as Fig. 33, but for cell 23.

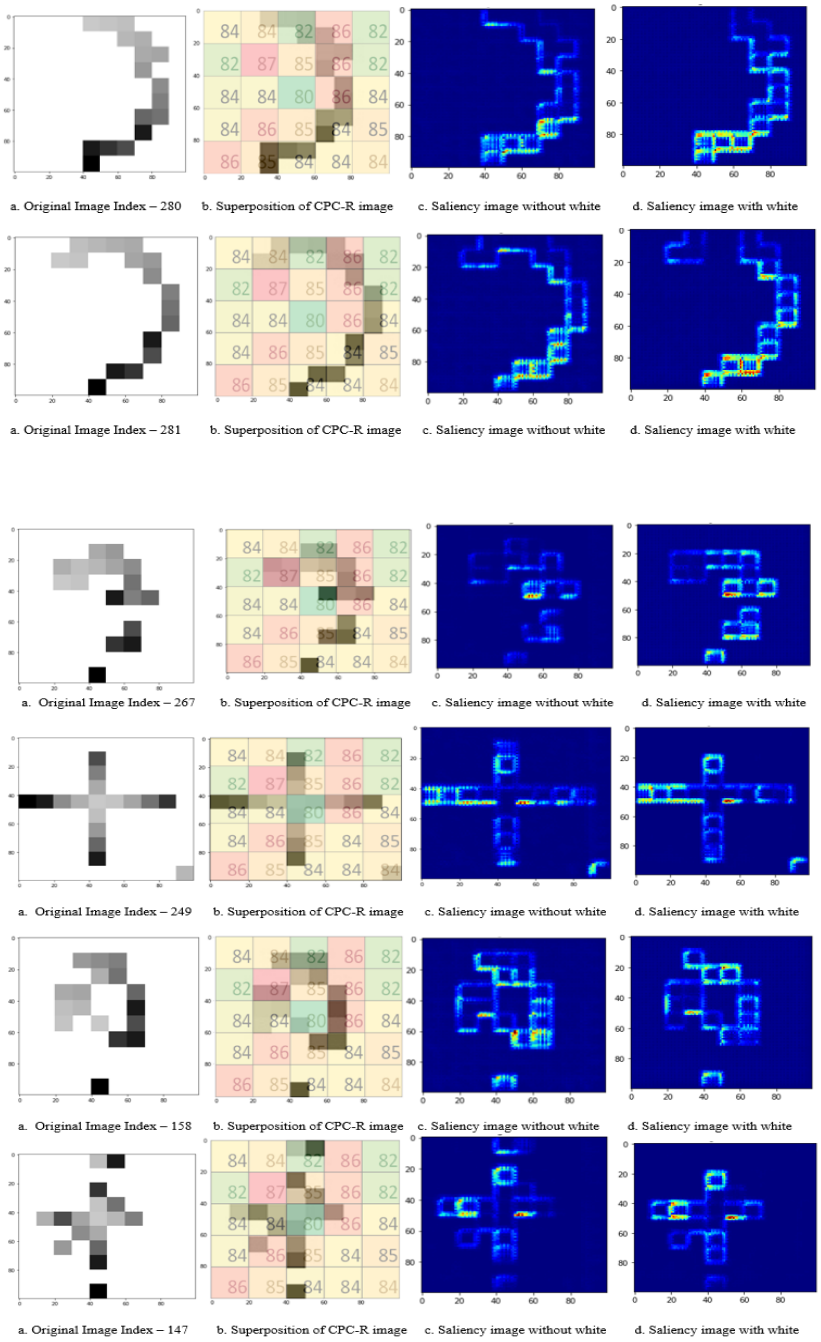


FIGURE 34: Informative cell 23.

The highest accuracy for the ICC method is 86.62% which is cell 17, which is supposed to be the least informative cell. The images in Fig. 35 present four types of images for this cell.

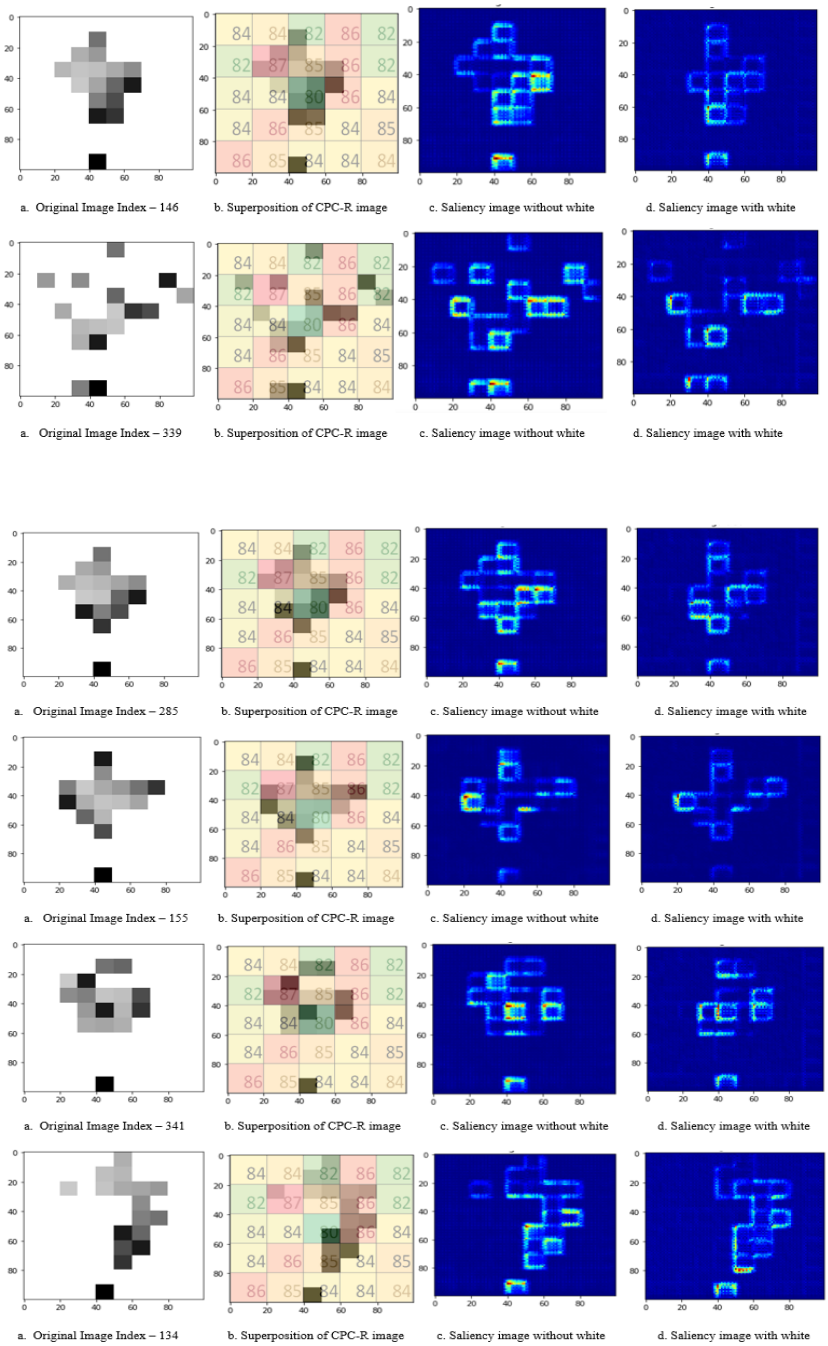


FIGURE 35: Informative cell 17.

All rows in Figs. 33-35 show that most Guided BackPropagation salient pixels (red) are well matched with the darkest boxes/pairs (x_i, x_j), i.e., pairs that are close to the start of all pairs. In this schema, the earlier pairs are encoded to be darker starting from black to very light gray in the gray-scale CPC-R images. Thus, this is not a very informative pattern, but rather an artifact of the CPC-R coding schema.

In contrast, Figs. 33-35 (b) show that cells 13, 23, and 17 contain pairs from CPC-R images that are present in Figs. 33-35 (a) not being the darkest ones, i.e., not Guided BackPropagation most salient in these images. This illustrates that the ICC algorithm discovers cells that are more relevant than Guided BackPropagation in CPC-R images for classification success.

Informative Cell Covering (ICC) with Glass data

Table 18 shows the accuracy of the classification of Glass data with covered cells. These cells are presented in the ascending order of accuracy.

TABLE 18: The classification accuracy of Glass data with covered cells.

Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
18	100×100	60	CNN64	10-fold	86.35
13	100×100	60	CNN64	10-fold	86.37
23	100×100	60	CNN64	10-fold	87.01
17	100×100	60	CNN64	10-fold	87.34
16	100×100	60	CNN64	10-fold	87.55
21	100×100	60	CNN64	10-fold	87.69
25	100×100	60	CNN64	10-fold	88.07
6	100×100	60	CNN64	10-fold	88.16
3	100×100	60	CNN64	10-fold	88.29
5	100×100	60	CNN64	10-fold	88.56
9	100×100	60	CNN64	10-fold	89.01
4	100×100	60	CNN64	10-fold	89.18
15	100×100	60	CNN64	10-fold	89.54

Table 18 (Continued)

Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
22	100×100	60	CNN64	10-fold	89.54
10	100×100	60	CNN64	10-fold	89.68
2	100×100	60	CNN64	10-fold	89.99
8	100×100	60	CNN64	10-fold	90.06
1	100×100	60	CNN64	10-fold	90.58
24	100×100	60	CNN64	10-fold	90.89
11	100×100	60	CNN64	10-fold	90.89
12	100×100	60	CNN64	10-fold	91.09
20	100×100	60	CNN64	10-fold	91.19
19	100×100	60	CNN64	10-fold	91.27

As was pointed out (see Table 18) the lowest accuracy for this method is 86.35% for cell 18, which is the most informative cell. The accuracy ranges from 86.35% to 91.27% with still cell 18 is most informative and cell 19 is among the least informative.

CHAPTER VI

COLLIDING AND SPLITTING VALUES WITH DIFFERENT ORDERS

The purpose of this experiment E10 to get better accuracy results and without loss of information. This is the new adjacent cells method to represent colliding pairs. The order for filling the adjacent cells of this experiment E10 differs from the previous experiment E2. A new ***spiral filling*** order used in this experiment that we call spiral filling:

- $right \rightarrow down \rightarrow left \rightarrow up \rightarrow lowerright \rightarrow lowerleft \rightarrow upperright \rightarrow upperleft$

Figs. 36 illustrates the method used in E2, and Fig. 37 illustrates the new method with Table 19 resenting the accuracy results of the new method.

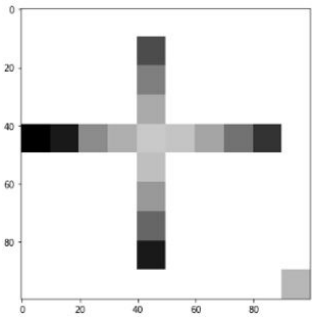


FIGURE 36: Adjacent cells with previous ***"cross"*** order.

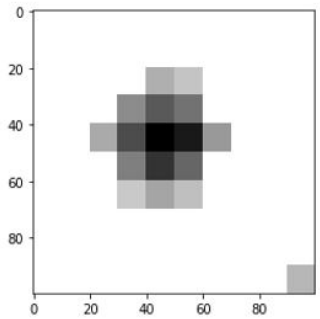


FIGURE 37: Adjacent cells with new ***"spiral"*** order.

TABLE 19: Results for E10 experiment with Ionosphere data.

Image Size	Epochs	Model	Cross validation	Accuracy
50×50	30	ResNetV2	10-fold	95.47
50×50	50	CNN64	10-fold	94.03
50×50	100	ResNetV2	Stratified 10-fold	93.46
100×100	30	CNN64	Stratified 10-fold	92.61
100×100	50	MLP	10-fold	91.91

Experiment E10 provided a better accuracy of 95.47% in comparison with the best results of the previous adjacent cells experiment (89.98%). Table 20 presents the results of cell accuracy for each covered cell that is made white and new adjacent order in this experiment E10. There is not much difference between the accuracies for the cells in this new adjacent order. The accuracy ranges from 88.21% to 91.15% with still cell 13 is most informative and cell 17 is among the least informative. The cells in the ascending order by accuracy values are shown in Table 20:

TABLE 20: Informative cells (ICC) results for Ionosphere data with spiral order.

Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
13	50×50	30	CNN64	10-fold	88.21
18	50×50	30	CNN64	10-fold	88.32
2	50×50	30	CNN64	10-fold	88.43
11	50×50	30	CNN64	10-fold	88.46
14	50×50	30	CNN64	10-fold	88.57
21	50×50	30	CNN64	10-fold	88.75
22	50×50	30	CNN64	10-fold	89.06
23	50×50	30	CNN64	10-fold	89.16
12	50×50	30	CNN64	10-fold	89.18
16	50×50	30	CNN64	10-fold	89.19
25	50×50	30	CNN64	10-fold	89.28
3	50×50	30	CNN64	10-fold	89.32
4	50×50	30	CNN64	10-fold	89.61
6	50×50	30	CNN64	10-fold	89.61
7	50×50	30	CNN64	10-fold	89.61
8	50×50	30	CNN64	10-fold	89.61
15	50×50	30	CNN64	10-fold	89.64

Table 20 (Continued)

Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
19	50×50	30	CNN64	10-fold	89.75
20	50×50	30	CNN64	10-fold	89.75
9	50×50	30	CNN64	10-fold	89.79
10	50×50	30	CNN64	10-fold	89.8
17	50×50	30	CNN64	10-fold	89.8
5	50×50	30	CNN64	10-fold	89.91
24	50×50	30	CNN64	10-fold	90.08
1	50×50	30	CNN64	10-fold	91.15

As was pointed out (see Table 20) the lowest accuracy for this method is 88.21% for cell 13, which is the most informative cell.

Multiple experiments have been conducted to explore the impact of combinations of properties of images on the accuracy of classification. Table 21 summarizes the results. It contains only results with the different order that produced the best accuracies.

TABLE 21: E3-E8 experiments result for Ionosphere data with new spiral order.

Experiments	Image Size	Epochs	Model	Cross Validation	Accuracy
E3	50×50	50	LeNet5	10-fold	93.85
E3	50×50	50	CNN64	10-fold	93.15
E4	50×50	50	CNN64	10-fold	93.98
E5	50×50	50	CNN64	10-fold	96.02
E5	50×50	50	LeNet5	10-fold	95.85
E6	50×50	50	LeNet5	10-fold	94.54
E6	50×50	50	CNN64	10-fold	93.78
E7	50×50	50	LeNet5	10-fold	92.98
E8	50×50	50	LeNet5	10-fold	95.98
E8	50×50	50	CNN64	10-fold	94.56

Table 21 shows the best accuracy results with the spiral order. These experiments are with the CPC-R images resulted in accuracies between 89.98% and 96.02% for the Ionosphere data. The achieved accuracy for the Ionosphere data is 96.02% in the E5 experiment. These experiments are with the MLP classifier provided lower accuracies

than other classifications. These experiments involve both optimizations of pairs in the order of the coordinates and their intensities.

Frequency for Informative Cells

Table 22 based on Table 20 shows the results for the analysis of cell 13 in Ionosphere data, which includes boxes (5,5), (5,6), (6,5), and (6,6) in E10. It shows that the most informative is pair (x_5,x_6) that is appeared 48 times in cell 13. The next most informative pair is (x_7,x_8) that is appeared 40 times in cell 13. The pairs (x_1,x_2) and (x_3,x_4) follow them with frequency 38, and so on. Not only these pairs are most informative but their normalized values 5 and 6 are the most informative. The actual original values of the normalized values 5 and 6 are in [0, 0.2) and [0.2, 0.4), respectively.

TABLE 22: Ionosphere data frequency for cell 13 with the "spiral" order of filling adjacent cells.

Pairs	(x_5,x_5)	(x_5,x_6)	(x_6,x_5)	(x_6,x_6)	Total
x_5,x_6	18	8	19	3	48
x_7,x_8	2	15	19	4	40
x_1,x_2	38	0	0	0	38
x_3,x_4	20	0	15	3	38
x_9,x_{10}	5	12	15	3	35
x_{11},x_{12}	2	12	8	6	28
x_{13},x_{14}	2	10	6	8	26
x_{33},x_{34}	5	4	7	10	26
x_{25},x_{26}	4	3	7	11	25
x_{29},x_{30}	5	3	8	7	23
x_{31},x_{32}	4	5	8	6	23
x_{17},x_{18}	3	4	5	9	21
x_{23},x_{24}	2	2	6	7	17
x_{21},x_{22}	1	2	4	7	14
x_{15},x_{16}	4	0	6	2	12
x_{19},x_{20}	0	3	2	6	11
x_{27},x_{28}	2	1	3	4	10

Table 23 contains the total frequency results for cell 14, cell 18, and cell 2 which are the next most informative cells.

TABLE 23: Frequency results for cell 14, cell 18, and cell 2.

Pairs	Cell 14	Cell 18	Cell 2
x_1,x_2	0	0	0
x_3,x_4	34	6	0
x_5,x_6	24	3	2
x_7,x_8	42	2	1
x_9,x_{10}	39	8	18
x_{11},x_{12}	34	10	17
x_{13},x_{14}	29	11	7
x_{15},x_{16}	55	7	7
x_{17},x_{18}	27	15	8
x_{19},x_{20}	38	14	4
x_{21},x_{22}	39	20	7
x_{23},x_{24}	24	19	7
x_{25},x_{26}	40	20	5
x_{27},x_{28}	34	12	3
x_{29},x_{30}	24	26	3
x_{31},x_{32}	20	27	2
x_{33},x_{34}	23	43	4

– Cell 14 consists of boxes (7,5),(5,6), (8,8), (8,6) with the most frequent informative pair (x_{15},x_{16}) that is appeared 55 times in cell 14. The next most informative pair is (x_7,x_8) that is appeared 42 times in cell 14. The pairs (x_9,x_{10}) and (x_{21},x_{22}) follow them with frequency 39, and so on. The actual original values of normalized values 7 and 8 are in [0.4, 0.6) and [0.6, 0.8), respectively.

– Cell 18 consists of boxes (5,3),(5,4), (6,3), (6,4) with the most informative pair (x_{33},x_{34}) that is appeared 43 times in cell 18. The next most informative pair is (x_{31},x_{32}) that is appeared 27 times in cell 18. The next most informative pair is (x_{29},x_{30}) that is appeared 26 times in cell 18. The pairs (x_{25},x_{26}) and (x_{21},x_{22})

follow them with frequency 20, and so on. The actual original values of normalized values 3 and 4 are in [-0.4, -0.2) and [-0.2, 0), respectively.

- Cell 2 consists of boxes (3, 9), (3,10), (4, 9), (4,10) with the most informative pair (x_9,x_{10}) that is appeared 18 times in cell 2. The next most informative pair is (x_{11},x_{12}) that is appeared 17 times in cell 2. The next most informative pair is (x_{17},x_{18}) that is appeared 8 times in cell 2. The pairs (x_{13},x_{14}), (x_{15},x_{16}), (x_{21},x_{22}), and (x_{23},x_{24}) follow them with frequency 39, and so on. The actual original values of this normalized value 7 are in [0.4, 0.6). The actual original values of this normalized value 8 are in [0.6, 0.8).

In summary, the results of this analysis show that most informative pairs of attributes are (x_{15},x_{16}) with frequency 55 and values in [7,8], pair (x_5,x_6) with frequency 48 and values in [5,6], and pair (x_{33},x_{34}) with frequency 43 times and values in [3,4]. In contrast with the traditional way, this algorithm shows informative pairs of attributes are not informative individual attributes. It highlights the mutual dependence of the attributes and their joint impact on classification accuracy.

CHAPTER VII

EXPERIMENTS WITH CAR DATA

These experiments have been conducted in the same setting as Ionosphere data and Glass data experiments. The Car data set includes 1728 instances of 4 classes with 6 attributes. The data has been normalized with [0,10] interval. Multiple experiments have been conducted to explore the impact of combinations of properties of images on the accuracy of classification.

These experiments (E1-E7) are conducted with the Car [25] data using MLP and CNN classifiers. Table 24 summarizes only results with the cross order and E1 with overlapping pairs that produced the best accuracies. All of them show 10-fold cross validation for accuracy on the validation data.

TABLE 24: Results of E1-E7 experiments for Car data with *cross* order.

Experiments	Image Size	Epochs	Model	Cross Validation	Accuracy
E1	50×50	100	CNN64	10-fold	94.79
E1	100×100	100	MLP	10-fold	83.68
E2	100×100	100	LeNet5	10-fold	91.01
E2	50×50	100	CNN64	10-fold	90.60
E2	100×100	100	MLP	10-fold	80.01
E3	50×50	150	LeNet5	10-fold	96.75
E3	100×100	100	CNN64	10-fold	93.99
E3	100×100	100	MLP	10-fold	87.09
E4	100×100	100	CNN64	10-fold	96.28
E4	50×50	100	LeNet5	10-fold	94.63
E4	100×100	100	MLP	10-fold	88.97
E5	100×100	150	CNN64	10-fold	96.8
E5	50×50	50	LeNet5	10-fold	92.03
E5	100×100	100	MLP	10-fold	83.87
E6	50×50	50	LeNet5	10-fold	92.04
E6	50×50	50	CNN64	10-fold	94.08
E6	100×100	100	MLP	10-fold	85.87

Table 24 (Continued)

Experiments	Image Size	Epochs	Model	Cross Validation	Accuracy
E7	50×50	100	CNN64	10-fold	94.07
E7	50×50	50	LeNet5	10-fold	96.28
E7	100×100	100	MLP	10-fold	87.91

Table 24 results are the best accuracy that produced with the cross order. These experiments are with the CPC-R images resulted in accuracies between 79.68% and 96.8% for the Car data. The achieved accuracy for the Car data is 96.8% in the E5 experiment. These experiments are with the MLP classifier provided lower accuracies than other classifications. These experiments involve both optimizations of pairs in the order of the coordinates and their intensities.

Informative Cell Covering (ICC) with Car data

In contrast with Table 24, Table 25 shows the accuracy of classification of Car data with covered cells. These cells are presented in the ascending order of accuracy.

TABLE 25: The classification accuracy of Car data with covered cells.

Cells	Image Size	Epochs	Model	Cross Validation	Accuracy
13	100×100	60	CNN64	10-fold	87.85
17	100×100	60	CNN64	10-fold	88.07
18	100×100	60	CNN64	10-fold	88.17
23	100×100	60	CNN64	10-fold	88.58
14	100×100	60	CNN64	10-fold	88.99
21	100×100	60	CNN64	10-fold	88.99
4	100×100	60	CNN64	10-fold	89.02
6	100×100	60	CNN64	10-fold	89.22
3	100×100	60	CNN64	10-fold	89.52
11	100×100	60	CNN64	10-fold	89.53
22	100×100	60	CNN64	10-fold	89.99
24	100×100	60	CNN64	10-fold	90.01
7	100×100	60	CNN64	10-fold	90.04

Table 25 (Continued)

16	100×100	60	CNN64	10-fold	90.58
10	100×100	60	CNN64	10-fold	90.98
12	100×100	60	CNN64	10-fold	90.93
8	100×100	60	CNN64	10-fold	91.65
9	100×100	60	CNN64	10-fold	91.66
25	100×100	60	CNN64	10-fold	91.88
19	100×100	60	CNN64	10-fold	91.91
20	100×100	60	CNN64	10-fold	92.11
1	100×100	60	CNN64	10-fold	92.11
5	100×100	60	CNN64	10-fold	92.11
2	100×100	60	CNN64	10-fold	92.11
15	100×100	60	CNN64	10-fold	93.68

As was pointed out (see Table 25) the lowest accuracy for this method is 87.85% for cell 13, which is the most informative cell. The accuracy ranges from 87.85% to 93.68% with still cell 13 is most informative and cell 15 is among the least informative.

CHAPTER VIII

COMPARISONS WITH OTHER STUDIES

The exact comparison of accuracies of different methods is difficult due to the multiplicity of the ways to test the results. Even when all accuracies are obtained in 10-fold cross validation of accuracies, the comparison is not exact, because different authors make different random splits of data into these 10 folds, put different time limitations on model optimization, and can use other unreported properties of the testing process. Therefore, we refrain from direct comparison with results, which are not in the 10-fold Cross Validation (CV) of accuracies that we used.

In our experiments, the best of accuracies for all data sets involve optimization of (1) pairing coordinates, (2) ordering these pairs, and (3) values of intensities of cells that encode them. The ResNet2 classifier is the winner that is simplest among the explored network architectures. Table 26 shows the comparisons of the accuracies of models presented in this paper with other classification models.

TABLE 26: Comparisons of different classification models.

Classification Algorithm	Accuracy
Breast Cancer Data:	
Visualization for solving non-image problems and saliency mapping (CPC-R with Cross order)	**95.58**
Visualization for solving non-image problems and saliency mapping (CPC-R with Spiral order)	**97.89**
GLC-R_5[Agrawal B. and Kovalerchuk B. 2019][1]	95.61
Deep Learning in Mammography[Becker S., Marcon M., Ghafoor S., Wurnig C., Frauenfelder T., Boss A. 2017][26]	94
DWS-MKL[Tingting W., Huayou S., and Junbao L., 2020][27]	96.9
LMDT Algorithm [Peter W., and Anh H.][28]	95.74

Ionosphere Data:	
Visualization for solving non-image problems and saliency mapping (CPC-R with Cross order)	**95.01**
Visualization for solving non-image problems and saliency mapping (CPC-R with Spiral order)	**96.02**
DWS-MKL[Tingting W., Huayou S., and Junbao L., 2020][27]	92.3
ITI Algorithm[Peter W., and Anh H.][28]	93.65
Deep Extreme Learning Machine and Its Application in EEG Classification[Shifei D., Nan Z., Xinzheng X., Lili G., Jian Z., 2015][29]	94.74
Glass Data:	
Visualization for solving non-image problems and saliency mapping(CPC-R with Cross order)	**96.80**
C4.5 Algorithm[Peter W., and Anh H.][28]	70.23
Glass Classification Using Artificial Neural Network/ANN Model[Mohmmad J., Bassem S., and Samy S., 2019][30]	96.7
Comparative Analysis of Classification Algorithms on Different Datasets using WEKA/MLP[Arora R., and Suman][31]	67.75
Car Data:	
Visualization for solving non-image problems and saliency mapping(CPC-R with Cross order)	**96.8**
Performance Comparison of Data Mining Algorithms[Jamilu A., Anahita G., Azuraliza A., 2014][32]	93.22
A Large-Scale Car Dataset for Fine-Grained Categorization and Verification[Linjie Y., Ping L., Chen C., and Xiaoou T., 2015][33]	83.22

Ionosphere data: For these data, 98.78% on training data and 100% on validation data were reported in [34] for the FSP algorithm that used another visual representation of n-D points as graphs and logical rules, not heatmaps and DL. It was done by single random splits of the data with 70% to training and 30% to validation data. Therefore, this result is not applicable for the direct comparison. The same paper summarized 10-fold CV from different sources: 93% for MLP, 94.87% for C4.5, 94.59% for Rule Induction, 97.33% for SVM without converting n-D data to images. Our achieved accuracy of 96.02% in this range. This means that CPC-R can be considered as competitive with these

algorithms in accuracy. The advantage of CPC-R is that it can be used with any DL and MLP algorithm uniformly with lossless n-D data visualization.

Glass data: Multiple results are reported from 68.2 for C4.5 to 98.13 for the random forest but without telling the accuracy evaluation method [35][4], or use 10-fold CV with the area under the ROC curve (AUC) and F [36], not accuracy. Therefore, it is hard to compare these results directly with our achieved 96.01% accuracy in 10-fold cross validation. Thus, our result is in the range of results reported in the literature and is competitive.

CHAPTER IX

CONCLUSION

This thesis presented CPC-R algorithms and results of computational experiments with CNN and MLP classifiers on images constructed from non-image data. These images were constructed by CPC-R algorithms and optimized the intensities and pairing of coordinates to improve the accuracy. The proposed CPC-R algorithm is a competing alternative to other Machine Learning algorithms as conducted experiments had shown. These experiments are with the CPC-R images resulted in accuracies between 69.8% and 96.02% for Ionosphere data, from 84% to 96.8% for Glass data, and from 83.68% to 96.8% for Car data with different classifiers.

Experiments E1 and E2 reported above had shown better accuracy with Inception ResNetV2 classifier for Ionosphere data and Glass data. The achieved accuracy for the Ionosphere data is 96.02% in E5 with spiral order, Glass data is 96.86% in E1, and Car data is 96.8% in E5 with cross order, which means the CPC-R can be considered as competitive with other algorithms in accuracy. The MLP is a competitive classifier to other CNN classifiers for both Ionosphere data and Glass data. In experiment E3, the MLP provided better accuracy than others for Glass data. The experiments with ***spiral filling*** order produced better accuracy compared to cross order. The ***spiral filling*** order is the best order. In ***cross order***, experiment E5 is the best experiment compared to other experiments.

My future work is conducting more computational experiments in an attempt to get better accuracies based on new modifications of the algorithm such as the location of colliding pairs. Another future study is extracting the visual features CPC-R images from

CNN models are most responsible for the high accuracy [2]. This study will be supported by visualization.

The advantages of CPC-R algorithms are lossless visualization of n-D data, with the ability to add context to the visualization by overlaying the images of n-D points with the mean images of competing classes. The CPC-R algorithms open the opportunity for a visual explanation of the model by tracing back to the salient image pixels and visualizing them using a heatmap. The proposed approach also can be used for data anonymization, which is important for many machine learning tasks. It will convert numerical n-D data into the anonymized images.

REFERENCES CITED

[1] B. Agrawal. and B. Kovalerchuk., "Solving non-image learning problems by collocated visualization and deep learning algorithms," *CWU technical report*, 2019.

[2] B. Kovalerchuk, B. Agrawal, and D. Kalla, "Solving non-image learning problems by mapping to images," vol. Melbourne, Victoria, Australia, 7-11 Sept.2020,DOI 10.1109/IV51561.2020.00050.

[3] D. Dovhalets, B. Kovalerchuk., S. Vajda, and A. Răzvan, "Deep learning of 2-d images representing nd data in general line coordinates," in *International Symposium on Affective Science and Engineering ISASE2018*, pp. 1–6, Japan Society of Kansei Engineering, 2018.

[4] A. Sharma, E. Vans, D. Shigemizu, keith Boroevich, and T. Tsunoda, "Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific reports*, vol. 9, no. 1, pp. 1–7, 2019.

[5] D. Dovhalets and B. Kovalerchuk, "Constructing interactive visual classification, clustering and dimension reduction models for nd data," in *Informatics*, vol. 4, p. 23, Multidisciplinary Digital Publishing Institute, 2017.

[6] B. Kovalerchuk., *Visual knowledge discovery and machine learning*, vol. 144. Springer, 2018.

[7] A. Asuncion and D. Newman, *Ionosphere Data Set*, 2007. `https://archive.ics.uci.edu/ml/datasets/Ionosphere`.

[8] A. Asuncion and D. Newman, "Uci machine learning repository," 2007.

[9] V. Spiehler, *Glass Identification Data Set*, 1987. `https://archive.ics.uci.edu/ml/datasets/Glass+Identification`.

[10] M. Bohanec and B. Zupan, *Car Evaluation Data Set*, 1997. `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`.

[11] J. Brownlee, "How to configure the number of layers and nodes in a neural network," *Retrieved from Machine Learning Mastery*, 2018.

[12] "Multilayer perceptron (mlp) vs convolutional neural network in deep learning," *Data Science Bootcamp website*, 2018. `https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-\newlinein-deep-learning-c890f487a8f1`.

[13] L. Raghupathi, G. Laurent, F. François, M. Damien, M. Cani, and C. Christophe, "An intestinal surgery simulator: Real-time collision processing and visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 6, pp. 708–718, 2004.

[14] Kamat, V. R., Martinez, and J. C., "Interactive collision detection in three-dimensional visualizations of simulated construction operations," *Engineering with Computers*, vol. 23, no. 2, pp. 79–91, 2007.

[15] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[16] "Saliency maps," 2020. `https://www.geeksforgeeks.org/about/`.

[17] R. Kotikalapudi and contributors, "keras-vis." `https://github.com/raghakot/keras-vis`, 2017.

[18] A. Radhakrishna and S. Sabine, "Saliency detection for content-aware image resizing," in *2009 16th IEEE international conference on image processing (ICIP)*, pp. 1005–1008, IEEE, 2009.

[19] H. Seunghoon, Y. Tackgeun, K. Suha, and H. Bohyung, "Online tracking by learning discriminative saliency map with convolutional neural network," in *International conference on machine learning*, pp. 597–606, PMLR, 2015.

[20] R. S. Ramprasaath, C. Michael, D. Abhishek, V. Ramakrishna, P. Devi, and B. Dhruv, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[21] A. Schreiber, *Saliency Maps for Deep Learning – Vanilla Gradient*, 1997. `https://andrewschrbr.medium.com/saliency-maps-for-deep-learning-part-1-vanilla-gradient/-1d0665de3284`.

[22] K. Beomsu, S. Junghoon, J. Seunghyeon, K. Jamyoung, C. Jeongyeol, and T. J., "Why are saliency maps noisy? cause of and solution to noisy saliency maps," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4149–4157, IEEE, 2019.

[23] N. Ernst, "Saliency map," *Scholarpedia*, vol. 2, no. 8, p. 2675, 2007.

[24] H. Shengfeng, L. R. WH, L. Wenxi, Z. Huang, and Q. Yang, "Super cnn: A super pixel wise convolutional neural network for salient object detection," *International journal of computer vision*, vol. 115, no. 3, pp. 330–344, 2015.

[25] M. Bohanec and B. Zupan, "Uci machine learning repository: Car evaluation data set," 1997.

[26] A. Becker, M. Marcon, S. Ghafoor, M. Wurnig, T. Frauenfelder, and A. Boss, "Deep learning in mammography: diagnostic accuracy of a multipurpose image analysis software in the detection of breast cancer," *Investigative radiology*, vol. 52, no. 7, pp. 434–440, 2017.

[27] W. Tingting, L. Junbao, and S. Huayou, "Dws-mkl: Depth-width-scaling multiple kernel learning for data classification," *Neurocomputing*, vol. 411, pp. 455–467, 2020.

[28] P. Eklund and A. Hoang, "A performance survey of public domain supervised machine learning algorithms," *Australian Journal of Intelligent Information Systems. v9 i1*, pp. 1–47, 2002.

[29] S. Ding, N. Zhang, X. Xu, L. Guo, and J. Zhang, "Deep extreme learning machine and its application in eeg classification," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[30] M. J. E., B. S. A., and S. S. A., "Glass classification using artificial neural network," 2019.

[31] R. Arora, "Comparative analysis of classification algorithms on different datasets using weka," *International Journal of Computer Applications*, vol. 54, no. 13, 2012.

[32] J. Awwalu, A. Ghazvini, and A. Bakar, "Performance comparison of data mining algorithms: A case study on car evaluation dataset," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 13, no. 2, 2014.

[33] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3973–3981, 2015.

[34] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, "Accurate intelligible models with pairwise interactions," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–631, 2013.

[35] M. M. R. Khan, R. B. Arif, M. A. B. Siddique, and M. R. Oishe, "Study and observation of the variation of accuracies of knn, svm, lmnn, enn algorithms on eleven different datasets from uci machine learning repository," in *2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT)*, pp. 124–129, IEEE, 2018.

[36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[37] D. C. Kalla, "Visualization for solving non-image problems and saliency mapping." https://github.com/KallaDivyachandrika/VISUALIZATION-FOR-SOLVING-NON-IMAGE-PROBLEMS-AND-\\SALIENCY-MAPPING, 2007.

APPENDIX A

SOFTWARE DESCRIPTION

This program was coded as a Python Programming Language version Python 3.7.0. The python is downloaded from the Anaconda distribution and many libraries are installed in Anaconda Prompt. Used **Spyder** IDE and **Jupyter Notebook** which is open-source software for all experiments.

The code that trains and tests a machine learning model with Keras backed by Tensor-Flow in Python. The code is written in python and will be tested in CPU and GPU using the Knuth machine. Keras is an open-source neural network library that is written in Python. Here the Keras will work over the Tensor Flow. Tensor Flow is a free and open-source software library for machine learning. Tensor Flow is the back-end engine that is recommended usually among the other back-end engines. These experiments run on Knuth High-Performance Computer machine by using the necessary commands.

APPENDIX B

REPOSITORY

The Python code at GitHub can be made available on the GitHub repository [37]. The data sets are taken from the UCI Machine Learning Repository [8].